Compressed Learning for Time Series Classification

Yuh-Jye Lee*, Hsing-Kuo Pao†, Shueh-Han Shih†, Jing-Yao Lin† and Xin-Rong Chen†
* Department of Applied Mathematics
National Chiao Tung University, Hsinchu, Taiwan
Email: yuhjye@math.nctu.edu.tw
† Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology, Taipei, Taiwan
Emails: pao@mail.ntust.edu.tw, m10215091@mail.ntust.edu.tw, jyl.ntust@gmail.com, m10315041@mail.ntust.edu.tw

***Abstract*—The time series classification has been studied for various applications in the last decades. In the time series classification problem, we decide the class information based on a small piece of the time series inputs. In general, the approaches to time series classification can be categorized into three types, distance-based, model-based, and feature-based approaches. In this research, we focus on the feature-based methods, which represent time series as a set of characterized values. It is quite often the case that features generated by existing representation techniques are not transparent to domain experts and the feature that are selected for classification are not completely interpretable. We aim to propose a novel time series representation, called *Envelope* to solve the problem. The proposed supervised feature extraction method transforms time series into simple $1/0/{-1}$ values. A heuristic is introduced to determine the most appropriate representation which includes the features that are the best to discriminate data of different labels. Moreover, this new representation enjoys the characteristic of sparsity which is an essential property when we need to apply compressed sensing techniques. With this advantage, we can benefit from high transmission efficiency, the reduction of required storage and model complexity. We conduct a series of tests on various benchmark time series data to show the effectiveness of the proposed method. Other than the classification effectiveness, we demonstrate how to visualize the similarity between time series of the same and different kinds from the proposed *Envelope* method.**

***Index Terms*—compressed sensing; IoT; SVM; time series classification;**

I. Introduction

We propose a novel sparse representation for time series classification. With the sparsity, it is allowed to introduce compressed sensing [5] for time series classification. By having it, we gain the computational and communication advantages in IoT applications and others. The definition of IoT is that ubiquitous "things" with limited communication, weak computation and small memory/storage, are seamlessly interwoven with each other to form a network [1], [9]. These "things" read the environment around them such as the interesting targets in the environment or the status of their own continuously. After that, the raw data are integrated and transformed into information, possibly by some machine learning methods, and then we can utilize the information to proactively provide services according to different contexts and personal needs.

The data collected in the scenario of IoT applications are often in a time series format. As a consequence, various time series analysis techniques are helpful to deal with such data. Due to the aforementioned limitation and constraints of IoT framework the conventional data analysis methods and machine learning algorithms can not be applied to IoT data analysis directly for practical applications. The "simplification" and the "approximation" principles from the conventional methods show one of the possible solutions to solve the puzzles. Compressed sensing [5] is regarded as a promising solution to many real applications which attracts great attention recently. This technique dramatically reduces the volume and dimensionality of original data and it provides a way for us to recover data with less information loss with high possibility. In addition, theoretical foundation [4] suggests that it is feasible to perform machine learning tasks in the compressed domain and reduce the computational complexity in this case. Nevertheless, it is applicable only when the input data own the sparse property, which is not the case for most real-world data. Constructing a sparse representation for the raw data becomes one of the most important steps before we apply compressed sensing to tackle real-world problems.

We focus on the *sparse representation* and *dimension reduction* for time-series data. In order to apply compressed sensing to IoT data analytics, we develop a novel sparse representation method for time-series data which uses an *envelope* to encode a set of time series if they own similar patterns. This envelope describes the "profile" for a set of time series. Given the profile, we encode an observation, to be either 0 or ± 1 depending on whether the observation falls inside or outside the envelope respectively on each moment. This encoding scheme also offers a *similarity measure* between a single time series and a set of time series. The higher sparsity indicates the higher similarity. Overall, the representation gives us a chance to utilize compressed sensing in our data analysis tasks.

The size of the envelope is very crucial to the performance of times series profiling and time series classification. To deal with the classification problems, we propose a heuristics to determine k (the size of the envelope) for each time-series type to make each type, therefore each envelope distinguishable from other types of time series or envelopes. Given the envelopes, we then represent each time series in a sparse format, a series of -1, 0 or $+1$. Numerical results demonstrate that we can apply the compressed sensing techniques to the encoded time series and train a classifier in the compressed

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

domain without sacrificing much the classification accuracy. Learning in the compressed domain gives us the advantages in both the computation and communication complexity. It reduces the power consumption and is suitable for small-scale IoT devices. The limitation of the proposed method is that it is only applicable to synchronized time-series data, such as machine logs with timestamps, or other datasets that can be synchronized easily. For the time-series data where the synchronization is hard to apply, we rely on, for example, any efficient alignment algorithm to match between different series data and it is not the main focus of this work.

Time series classification approaches can be categorized into three types [14]: (1) the feature-based approach, transforming time series with predefined features [13], [11], and applying conventional classifiers; (2) the distance-based approach, with kNN to be the state-of-art algorithm with dynamic time wrapping [10], and having been proved to be robust and difficult to beat; (3) the model-based approach, building a model for data within a certain class, and calculating the likelihood with the model, for instance, hidden Markov model (HMM). The key to handle time series with effectiveness and high efficiency is to choose an appropriate representation method for the time-series data [12]. In this article, we focus on feature-based time-series classification [8]. More precisely, we are interested in time-series representation/dimensionality reduction techniques. There is always a need to transform time series into a more concise format since the cost of computation, storage and time can be reduced. Time series is a fundamentally high dimensional data, which means directly dealing with time series raw data is unwise. Therefore, we introduce a new time-series representation method called *Envelope*, which measures a set of statistical features from given training data, then transforms incoming time series into their sparse format.

This novel representation not only can be combined with the compressed sensing techniques for learning, but can also provide interpretable results and competitive performance. In addition, it can be employed to solve anomaly detection tasks. As shown in experiments, the proposed representation technique is robust to noise and has excellent compressibility, which makes it favorable when deploying the idea to real-world scenarios. The rest of this paper is organized as follows. Section II provides a brief introduction of compressed sensing. In Section III, we introduce the proposed method *Envelope* in details. In Section IV, a classification framework of the proposed method will be addressed. Section V delivers the experiments to verify the effectiveness, robustness and efficiency of the proposed method, which is followed by Section VI, where we apply the proposed method to real-world applications. We draw the conclusion Section VII in the end.

II. Compressed Sensing

Compressed sensing is a novel and effective data sensing (sampling) technique which dramatically reduces the storage and transmission cost while it can recover a signal from compressed domain perfectly. Compressed sensing theory is popular since it needs much fewer number of measurements than traditional techniques.

A sparse signal $x \in \mathbb{R}^n$ can be compressed via a linear transformation, such as $Ax = y$. The matrix $A \in \mathbb{R}^{n\times p}$ is called a *measurement* or *sensing* matrix which compresses a signal from n dimension to a much lower dimension p and each row is regarded as a *measure* of the sparse signal x. The goal of compressed sensing is keeping the number of measures as few as possible while the recovery ability remains.

According to the fundamental linear algebra,

$$Ax = y,\ A \in \mathbb{R}^{p\times n},\ \text{and}\ p \ll n\,,$$

this linear system of equations shall not have a unique solution. That is, it is impossible to have a one-to-one mapping between x and y unless we add on assumptions of our interests x and restrict the sensing matrix A. First, we assume that we are only interested in the sparse solutions of the linear system. That is, x is k-sparse vector, which means that there are at most k nonzero values in x. Usually k is much smaller than n. Moreover, matrix A satisfies the *Restricted Isometry Property* (RIP) of order m if there is some constant, $\delta_k \in (0, 1)$, s.t.

$$(1-\delta_k)\|x\|_2^2 \le \|Ax\|_2^2 \le (1+\delta_k)\|x\|_2^2\,,\ \forall\ k\text{-sparse vector } x\,. \tag{1}$$

This means that the *length* of a vector Ax in the compressed domain will be very close to the *length* of vector x in the data domain. Given the sparsity assumption of x, we can recover the original data x from compressed domain, $y = Ax$ without information loss. It can be achieved by solving an L_1-norm minimization problem that can be formulated as a linear programming problem. It is an expensive step in compressed sensing. However, if we are only interested in the *data pattern*, saying classification models, we probably can skip this expensive step. That is analyzing the data in the compressed domain. Similar idea can be found in [4]. It proves that SVM (support vector machine) can keep its learnability in compressed domain. That is, if the data is approximately linear separable in high-dimensional data domain, then the SVM can roughly keep the linear separable property in the compressed domain. In some cases, if the task we are dealing with is not that complicated or difficult, then it is suitable to handle this task directly in the compressed domain for efficiency.

There are some studies [2], [7], [3] that claimed that the choice of measurement matrix A can be a normal random matrix generated with specific parameters ($\mu = 0, \sigma = \frac{1}{\sqrt{p}}$), which will have high probability to satisfy the RIP and is good enough for most real-world applications. From the machine learning point of view, its characteristic reduces the model complexity and diminishes the "curse of dimensionality". In order to enjoy these advantages, we introduce a novel sparse representation for time series so that we can tackle IoT data analytics issues.

III. Sparse Envelope Representation for Time Series Dataset

The idea of *envelope* has been utilized in finance field for a long time. According to INVESTOPEDIA (http://

www.investopedia.com/), *envelope* is defined as follow: An indicator is formed by evaluating the two moving averages, which defines the upper and lower price range boundaries. It is used by investors and traders to help analyze extreme overbought and oversold conditions in a market. Motivated by this observation, we try to apply such idea to time series representation.

Time series data are the major type of IoT data. In order to handle transmission and storage issues in IoT scenarios, we try to find a sparse representation for time series data so that it is possible to apply compressed sensing for efficiency. In this research, we consider a set of well synchronized, same-length time series data which comes from the same source and we attempt to discover a novel sparse representation for it. A time series can be represented as a vector in its temporal order:

$$T \;=\; (t_1, t_2, \ldots, t_n),\; t_j \in \mathbb{R}\,, \tag{2}$$

where t_j is the value at timestamp j; and a time series dataset is given by:

$$D = \{T^i \mid T^i = (t_1^i, t_2^i, \ldots, t_n^i),\; i = 1 \ldots m\}\,, \tag{3}$$

where T^i stands for the i^{th} time series instance of D with length n; t_j^i represents its values at the j^{th} timestamp. We construct an *envelope* that is capable to characterize a time series dataset D. Since the time series in D are well-synchronized with the same length, we treat the j^{th} timestamp outcomes as samples generated by random variable $\mathbf{T}_j$. Then D can be regarded as a set of samples drawn from random variables $\{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_j, \ldots, \mathbf{T}_n\}$. The sample mean and sample standard deviation of random variable $\mathbf{T}_j$ is μ_j and σ_j respectively. Now, we define the *envelope* with size k, E_k, for dataset D as:

$$\begin{aligned} E_k = \{Z \mid\; & Z = (z_1, z_2, \ldots, z_n)\,, \\ & |z_j - \mu_j| \le k\sigma_j,\, Z \in D, z_j \in \mathbb{R}\}\,, \end{aligned} \tag{4}$$

where Z is any time series instances. The meaning of this definition is that using the values which are not too far away from the mean values to describe the dataset. Then we can use this *envelope*, E_k, to encode a time series T as a series $S = (s_1, s_2, \ldots, s_j, \ldots, s_n)$ consisting of -1, 0 and 1, where for all t_j in T, we define:

$$\begin{cases} s_j = 1, & \text{if} \quad t_j > \mu_j + k\sigma_j \\ s_j = -1, & \text{if} \quad t_j < \mu_j - k\sigma_j \\ s_j = 0, & \quad\;\;\, \text{otherwise}\,. \end{cases} \tag{5}$$

We illustrate the proposed idea in Figure 1 with the ECG Five Days dataset from UCR. Several time series instances are visualized on the top; the *envelope* with size $k = 2$ and corresponding mean curve are shown on the bottom. Obviously, most of the time series instances are covered by the *envelope* (shadowed area). In other words, the *envelope* can profile this time series dataset well. Besides, the series encoded by the *envelope* will be sparse series, which is an important requirement for compressed sensing. The detail implementation of the proposed *Envelope* is provided in algorithms 1 & 2.

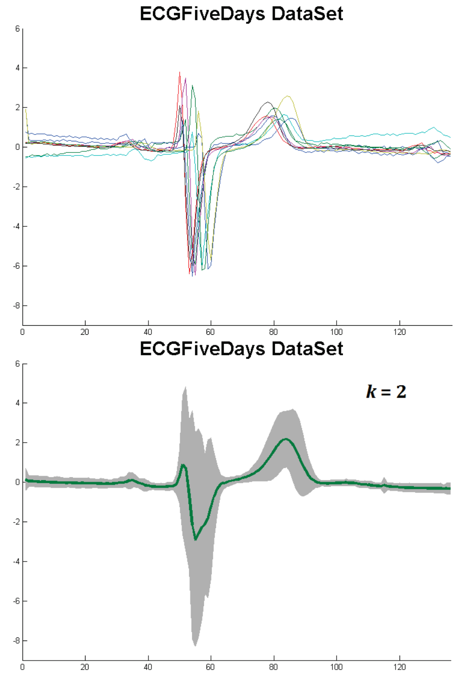


Fig. 1: Illustration of raw data and *envelope* of ECGFiveDays dataset.

Algorithm 1 $EnvelopeBuild(\mathrm{data}, \mathrm{label})$: The pseudo code for generating the models for transforming incoming series.

```
// data - the instances for building envelopes
// label - correspond to data instances
// meanCurve, stdCurve -
// matrices with size (# of Class × Series length)
for each unique class in label do
   Initialize μ, σ
   for each time stamp do
      Calculate μ_j and σ_j for the current class
      Append the result to μ, σ
   end for
   Append μ/σ to (meanCurve/stdCurve)
end for
return  meanCurve, stdCurve
```

According to Chebyshev's inequality

$$Pr(|X - \mu| \ge 1) = \frac{1}{k^2}\,. \tag{6}$$

The random variable X in the formula can be the random variable $\mathbf{T}_j$, for $i = 1, \ldots, n$. Based on the inequality, we can reasonably infer that the *envelope* representation owns the sparsity, and the statement is true for any distribution. That is, we expect sparse encoding result for $\mathbf{T}_j$ that follows any distributions. For $k = 2$, as an example, at least 75% observation values of time series at each timestamp will be set to 0. This kind of sparse representation implies the possibility of applying compressed sensing to obtain further boost in performance.

Algorithm 2 $EnvelopeEncode(\text{meanCurve}, \text{stdCurve}, T, k)$: The pseudo code for converting an instance to its sparse form.

```
// instance - time series needed to be transformed
// k - multiple of standard deviation
// encodingResult - the output sparse representation
Initialize encodingResult
for each unique class do
    Initialize temp as a vector of zeros
    If t_j of T > μ_j + kσ_j, set temp_j to 1
    If t_j of T < μ_j + kσ_j, set temp_j to -1
    Append temp to encodingResult
end for
return encodingResult
```

IV. Learning in the Compressed Domain

In this work, we mainly focus on the multi-class classification task given time series data. Apparently, the determination of k is a critical issue, which directly affects the efficiency of compressed sensing and effectiveness of *Envelope* representation. Hence we propose a heuristics to determine the size of *envelope*, k, in multi-class classification cases. Referring to Figure 2, the heuristics is further depicted as follows:

$$k^* = \underset{k}{\text{argmax}}(-a_k + \lambda b_k)\,, \tag{7}$$

where a_k is the non-zero ratio of target class given k; b_k is the gap of non-zero ratio between target class and the one in remaining classes with the smallest non-zero ratio at certain k. Here we define "target class" as the class that *envelope* is built with the instances from that class. Using the CincECG dataset from UCR as an example, we use data from class 1 to build *envelope* and encode all time series in the dataset, then the class 1 in this case is referred as the "target class". Additionally, the value of λ is predefined (e.g. 3 as the default value), a different λ may have a different effect on the compressibility of the proposed method. Afterwards, we evaluate all possible k (in our experiment, $k = 0 \sim 3$), and select the best one with the largest object value. In fact, the k in this heuristic is the tradeoff between the sparsity and the discriminative ability. On the other hand, employing the *envelope* to encode the target class (for example, using *envelope* built with class 1 instances to encode class 1 instances) returns a sparser result; while encoding time series from other classes should result a denser representation. Moreover, these results are interpretable since non-zero terms often occur at certain timestamps, indicating the key differences or patterns between classes. We notice that the encoding results are relatively sparse, meaning that it is possible to reduce the volume of storage. In Figure 3, (a) is the result of encoding class 1 data with *envelope* built by class 1; (b) is the encoding result of class 1 data with *envelope* built by class 2 and so on.

The *envelope* representation keeps local properties as well as makes converted time series discretized and interpretable. In addition, due to the sparse property mentioned above, we can further apply compressed sensing to it. From the machine learning viewpoint, managing the data with dimension reduction is always favorable. We are not only getting benefit from reduction in execution time but also diminishing the influence of the curse of dimensionality. In [4], the proof is provided that it is possible to perform SVM in compressed domain while keeps the same effectiveness. A theoretical bound was introduced to claim that data can be directly measured in the compressed domain.

An overall workflow is as follows: starting from time series

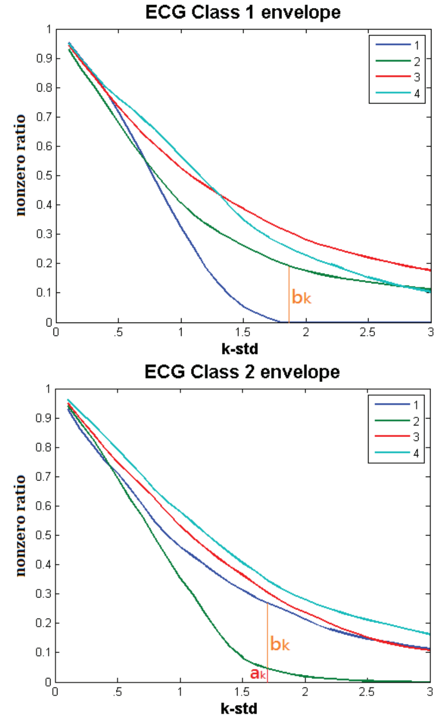


Fig. 2: The non-zero ratio for time series data respect to each class.

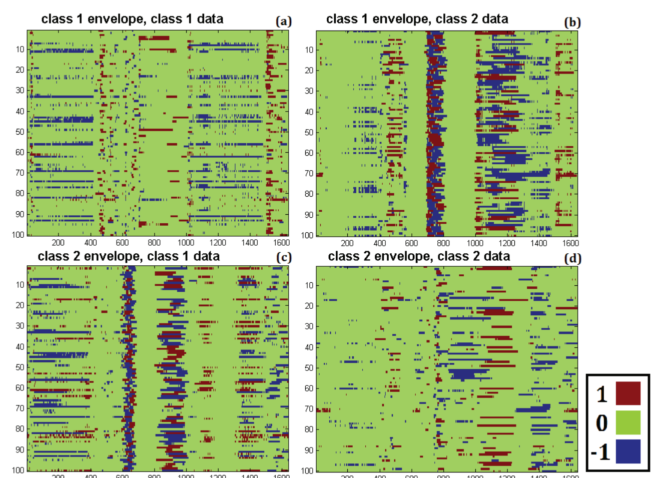


Fig. 3: The encoding result of ECGFiveDays dataset. The red is for 1, blue for -1 and green for 0.

of a certain class, *envelope* is built for that class beforehand. From acquisition of an incoming time series, encoding a series into its sparse version using the *envelope* with a given size k. After that, it is possible to transform the encoding result into a more compact form. What is worth mentioning is that the more compact an encoding result is, the less information is kept.

Next, the classification framework for the following experiments is provided (Figure 4). Starting from raw data, and go through data preprocessing to get the clean data. Then use this data as the input of *envelope* representation workflow, and we can get the *envelope* for each class with specific k provided by the heuristic that we proposed. Thus we can get the encoding result for each class and concatenate all of them to get the final representation. The encoding results of all classes can be concatenated together as features for linear SVM model training. In the next section, we use this framework to run the experiments without the preprocessing stage for fairness. However, not all of the datasets from the UCR archive are well-synchronized and de-noised. To obtain good enough performance in real-world applications, it is suggested to perform preprocessing for each task.

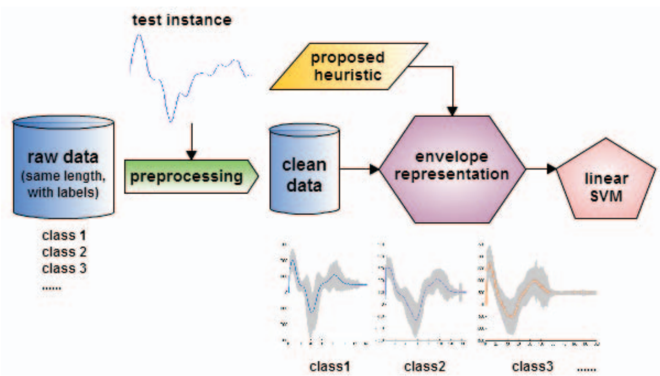


Fig. 4: Workflow of the proposed *Envelope* representation.

V. Numerical Result

In this section, the proposed classification framework is verified with a set of experiments. All the following experiments are implemented in Matlab, with Intel i5-3470 CPU, 16G RAM and Win 7 environment. All the parameters (λ for the heuristic, k for *envelope* and C for the linear SVM) are fine-tuned to achieve the best performance. The purpose of the experiments is to verify that the proposed method has the following features:

1) The proposed classification framework is competitive with the state-of-art methods.
2) *Envelope* representation is beneficial to computational efficiency.
3) *Envelope* representation is noise-resistant for tackling real-world signals.

A. Classification result

Our main object is to establish a novel sparse representation for compression while still keeps its distinguishability. To evaluate the *envelope* representation, we compare the performance of the proposed method with that from the state-of-art methods. We use the experimental settings similar to those from the data providers. Table I shows part of result of the whole UCR archive [6]. Obviously, the proposed method does not always outperform the state-of-art methods. Our claim is that, since most of datasets from UCR lack of training instances (near half of all datasets own fewer than 30 instances per class), it is difficult to get a stable *envelope* for classification. Hence the encoding result would have weak performance. However, it is not the case in the IoT scenario, which means the proposed method can still be effective for real-world applications.

TABLE I: Classification accuracy on UCR datasets

Dataset / Algorithm	1NN Euclidean	1NN DTW (best, noWin)	Envelope +SVM
Adiac	61.1	60.9/60.4	**67.7**
CBF	85.2	99.6/99.7	90.66
CinC ECG	89.7	93/65.1	58.2
Coffee	75	82.1/82.1	**85.71**
diatomSize Reduction	93.5	93.5/96.7	88.88
ECG 200	88	88/77	84
ECGFiveDays	79.7	79.7/76.8	**88.38**
Haptics	37	41.2/37.7	**43.5**
ItalyPower Demand	95.5	95.5/95	**97.08**
Lighting 2	75.4	86.9/86.9	77.04
Lighting 7	57.5	71.2/72.6	**75.34**
MALLAT	91.4	91.4/93.4	91.13
Mote strain	87.9	86.6/83.5	87.77
Olive oil	86.7	83.3/86.7	83.33
Sony	85.9	85.9/83.1	77.37
Sony II	69.5	69.5/72.5	**82.79**
Swedish leaf	78.7	84.3/79	**90.4**
Symbols	90	93.8/95	87.13
Synthetic control	88	98.3/99.3	94.66
Wafer	99.5	99.5/98	97.84

B. Effect of compression ratio

The main contribution of this research is the innovative representation, which can be combined with compressed sensing for efficiency. Hence, it is necessary to verify the impact of compression ratio on classification performance. Here, "compression ratio" is defined as the dimensions (in percentage) of *Envelope* representation remaining after we apply the data compression technique, or simply p/n (# of measurements divided by data dimension) from compressed sensing point of view. Figure 5 is an analysis about how the compression ratio of compressed sensing affects the classification performance. We select several datasets from UCR for demonstration.

Obviously, most datasets only have a little performance drop when the compression ratio reaches $1/8$ or even $1/16$, which shows that *envelope* representation has excellent compressibility. Besides, the compressed data get benefits from efficiency while keep competitive performance, which is very promising. However, the rest cases, the size can merely be reduced to about $1/4$ and the accuracy decreases significantly

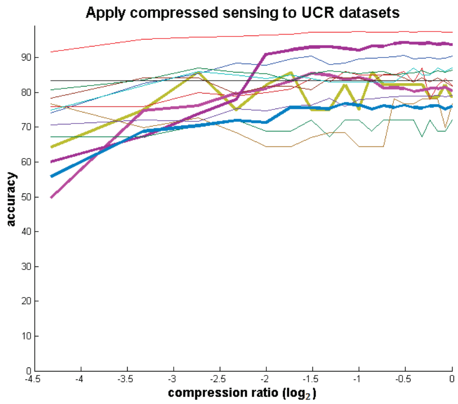


Fig. 5: Nearly $1/4$ of datasets from UCR collection are selected for the compression performance test.

when compression ratio decreases (bold lines). The detail is shown in Table II, we can found that the datasets with huge performance drop generally have shorter time series instance length. It is due to the fact that the dimension reduction techniques have their limitation, we can not expect a good performance for data that have only a few dimensions. Using an extreme case for explanation, compressing a time series from length 100 to 10 while keeping the performance is much easier than compressing from length 10 to 1 even they have the same compression ratio.

C. CPU time

A brief complexity analysis is discussed in this section, with the support from some experiments. Here, we set $|D|$ as m and $|T|$ as n for convenience. First, building *envelope* takes $O(m \cdot n)$, encoding each instance takes $O(n)$. In this paper, we use a linear SVM (from libSVM) as the classifier, it is expected to have an $O(m^2)$ complexity. In summary, the overall execution time from extracting features, tuning to testing is around $O(m \cdot n) \sim O(m^2)$, depends on the data set size.

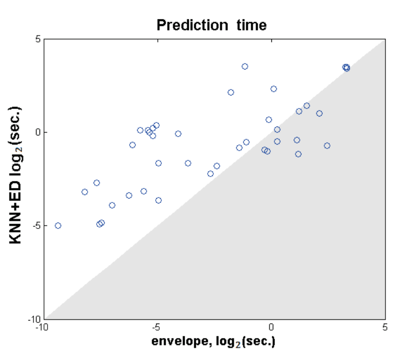


Fig. 6: The execution time on UCR datasets.

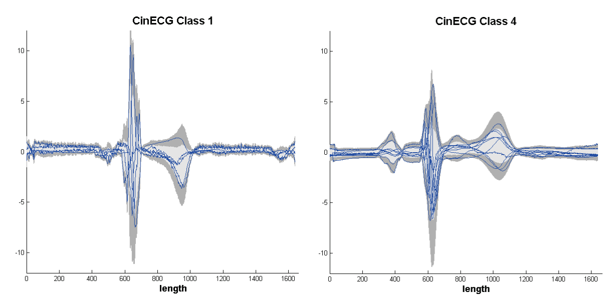
In the test phase, it only takes linear time to make prediction for incoming instances. As shown in Figure 6, the execution time of the proposed method is competitive to $1NN$ with Euclidean distance. Moreover, employing compressed sensing to the sparse instances could obtain extra boost in execution time. (The result shown above takes no compression techniques.)

D. Memory usage

Suppose that we have a data set $D_{m \cdot n}$, with m time series and n values for each instance. We assume that each value in a raw time series requires 32 bits (a conservative estimation). Through the *envelope* representation introduced above, it is possible to achieve 32 to (2 $\cdot NumberOfClass$) compression ratio. In addition, if a dataset can keep most of the information when it is compressed through compressed sensing, we can achieve 32 to (32 $\cdot NumberOfClass \cdot$ compression ratio) ratio of dimension reduction. Apparently, *Envelope* representation is not suitable for datasets with a large number of classes. In general, the proposed method has great space efficiency without compressed sensing. Moreover, if an encoded time series has low entropy, meaning that most data points in that sequence belong to a certain value. Then it is possible to further compress the data through *run-length* encoding.

E. Discussion

In this section, we use some datasets from the UCR archive as examples, hoping to figure out the reason why the proposed classification framework outperforms or loses the state-of-art methods. Then users can follow these steps to determine whether *envelope* is suitable for a specific task or it is needed to perform data preprocessing for further improvement.



(a) (b)

Fig. 7: There are only few instances in class 1; for time series in class 4, some time intervals have large variance so that the encoding result of these parts are information-less.

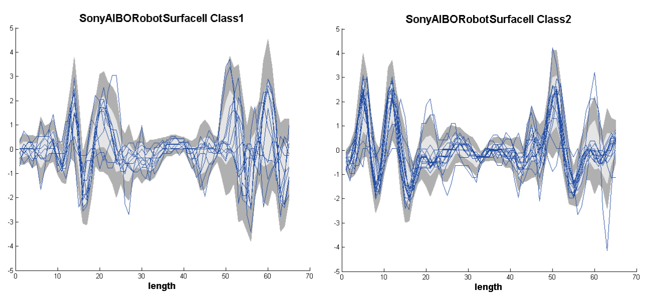
As mentioned in previous experiments, some datasets from the UCR archive lack of enough training instances, which leads to unstable *envelope* thus the unreliable encoding result, for example, the CinCECGtorso dataset (Figure 7). In average, there are 10 instances per class in the set. Additionally, some datasets are unbalanced and we find that some classes even have fewer than 5 instances. Due to this fact, users should check the number of instances per class before applying the proposed classification framework.

However, in some situations *Envelope* can have better performance than the state-of-art methods. For instance, the SonyAIBORobotSurface (Figure 8) dataset. We can see that there are some values exceeding the range of *envelope* (light gray: $k = 1$; dark gray: $k = 2$), indicating the noisiness of the dataset. When there exists noise in the data set, the proposed method usually outperforms the state-of-art methods. The

TABLE II: Classification accuracy on UCR datasets with different compression ratio

Dataset / Algorithm	Envelope+SVM 10% compression	Envelope+SVM 20% compression	Envelope+SVM 50% compression	Envelope +SVM
Adiac	70.58	69.56	69.82	67.7
CBF	85.88	86.88	91.11	90.66
CinC ECG	56.67	58.18	57.02	58.2
Coffee	85.71	89.28	85.71	85.71
diatomSizeReduction	95.75	95.42	96.07	88.88
ECG 200	84	83	81	84
ECGFiveDays	79.7	79.7	78.62	88.38
Haptics	43.83	43.5	43.5	43.5
ItalyPowerDemand	72.78	82.31	94.07	97.08
Lighting 2	63.93	70.49	63.93	77.04
Lighting 7	67.12	63.01	72.6	75.34
MALLAT	89.55	91.21	91.59	91.13
Mote strain	81.3	82.98	84.98	87.77
Olive oil	83.3	83.3	83.3	83.33
Sony	63.72	71.38	73.87	77.37
Sony II	74.92	79.43	78.59	82.79
Swedish leaf	85.76	87.68	91.04	90.4
Symbols	85.02	85.92	86.13	87.13
Synthetic control	82.67	86.33	92.67	94.66
Wafer	93.21	95.61	96.75	97.84

result also shows that distance-based time series classification methods are generally not robust to noises.



(a) (b)

Fig. 8: Distance-based methods have inferior performance on the datasets with noise.

VI. Case Study

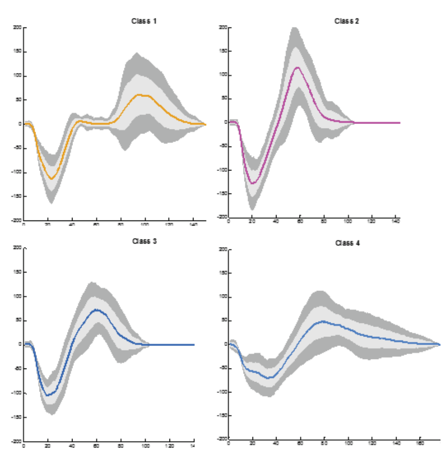
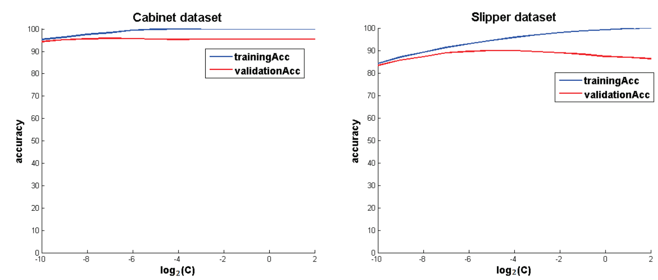


Fig. 9: The *Envelope* representation for each user on axis 5.



(a) (b)

Fig. 10: (a) The classification result for door-opening recognition, using axis 1 and 5; and (b) The classification result on the slipper dataset.

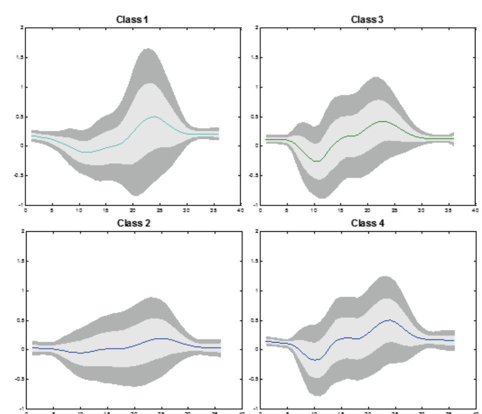


Fig. 11: Data collected through dynamic range checking.

We use a real-world dataset collected on our own to verify the proposed method that is truly effective for real applications. For a smart home scenario, sensors are applied to a home environment in order to provide services for home members. In this project, we attempt to identify home members through door-opening/closing and slipper gait recognition. The BLE sensor, EcoBT Mini (http://eco.epl.tw/hardware/ecoBTmini.html) is used with the sample rate 33Hz for data collection. Our goal is to recognize family members without asking users for further information and provide proactive services. We call this idea "passive user identification". Based on the data collected from each user's daily behavior, we are capable of identifying which family member the user belongs to. Thus the system can change the home environment settings according to the personal preferences.

For the first case, we deployed the sensors to cabinet doors[1]. The first three columns are the readings of accelerometer, and the following three columns are the readings of gyro meter. We use these trajectories to discover patterns for specific users and perform user identification. Each pair of opening/closing movement is regarded as a time series instance. The classification result is displayed in Figure 10(a). For a four class classification task, we can achieve about 90% accuracy, which shows that the proposed method is indeed effective on real-world cases.

For the gait recognition case, we install the sensors to slippers[2], collected in the same way as the dataset mentioned above. The result of recognition task is shown in Figure 10(b) and Figure 11 depicts the profiles of four walkers. In this four-class classification task, we are able to obtain over 80% accuracy, which demonstrates that the proposed method is also suitable for distinct cases. Besides, the experiment treats each step as an instance. In real-world applications, we can collect several steps then make a more robust prediction. For such case, we can reach more than 90% classification accuracy.

VII. Conclusion

In order to apply compressed sensing to IoT data analytics, we develop a novel sparse representation method for a set of time series which uses an *envelope* to encode the time series data. This *envelope* describes the *profile* for a set of time series. If an observation falls in the boundaries of the *envelope*, encode it by 0. Otherwise, encode it into either $+1$ or -1 when the observation falls above or below the *envelope* respectively. This encoding scheme gives us a measurement of similarity between a time series and a set of time series. The higher sparsity indicates the higher similarity. Obviously, the decision of k is very critical. For dealing with the classification problems, we propose a heuristics to determine k (the size of the *envelope*) for each category of time series to make each *envelope* distinguishable to other classes, then use these *envelopes* to represent original time series in a sparse format with -1, 0 and $+1$. Numerical results demonstrate that we can apply the compressed sensing to the encoded time series and train a classifier in the compressed domain without sacrificing much on the classification accuracy. Learning in the compressed domain gives us the advantages in both computation and communication aspects. Hence, it will reduce the power consumption, which is suitable for IoT scenarios. The limitation of the proposed method is that it is only applicable to synchronized time series data, such as machine logs with timestamps, or other datasets that can be synchronized easily. For more general time series, we have to develop the simple synchronization scheme, which will be addressed in future works.

[1]The dataset is available at http://dmlab8.csie.ntust.edu.tw/datasets/CabinetDataset.rar.

[2]The dataset is also available at http://dmlab8.csie.ntust.edu.tw/datasets/slipper_2.rar.

Acknowledgment

This work received grant from Ministry of Science and Technology of Taiwan under the grant number MOST 105-2119-M-001-035; also supported in part by the Ministry of Science and Technology of Taiwan (MOST 105-2633-E-002-001), National Taiwan University (NTU-105R104045), and Intel Corporation. The authors also acknowledge anonymous referees for their constructive criticisms.

References

[1] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Comput. Netw.*, 54(15):2787–2805, Oct. 2010.

[2] W. Bajwa, J. Haupt, G. M. Raz, S. Wright, and R. Nowak. Toeplitz-structured compressed sensing matrices. In *Statistical Signal Processing, 2007. SSP '07. IEEE/SP 14th Workshop on*, pages 294–298, Aug 2007.

[3] R. Baraniuk, M. Davenport, R. Devore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constr. Approx*, 2007.

[4] R. Calderbank, S. Jafarpour, and R. Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. Technical report, Princeton University, 2009.

[5] E. Candès and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[6] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. The UCR time series classification archive, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.

[7] A. Cohen, W. Dahmen, and R. Devore. Compressed sensing and best k-term approximation. *J. Amer. Math. Soc*, pages 211–231, 2009.

[8] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proc. VLDB Endow.*, 1(2):1542–1552, Aug. 2008.

[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.*, 29(7):1645–1660, Sept. 2013.

[10] E. Keogh and A. C. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2004.

[11] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, Mar. 2011.

[12] C. A. Ratanamahatana, E. J. Keogh, A. J. Bagnall, and S. Lonardi. A novel bit level time series representation with implication of similarity search and clustering. In *Advances in Knowledge Discovery and Data Mining, 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005, Proceedings*, pages 771–777, 2005.

[13] N. Ravi, N. D, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In *In Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence(IAAI*, pages 1541–1546. AAAI Press, 2005.

[14] Z. Xing, J. Pei, and E. Keogh. A brief survey on sequence classification. *SIGKDD Explor. Newsl.*, 12(1):40–48, Nov. 2010.