SCAFE: Automated simultaneous clustering and non-linear feature extraction of building energy profiles

Ulyanin, Stepan¹; Vazquez-Canteli, Jose²; Park, June Young²; Nagy, Zoltan²

¹Georgia Institute of Technology, Atlanta, GA, USA
²Cockrell School of Engineering, The University of Texas at Austin, TX, USA

E-mail: nagy@utexas.edu

Abstract. Increasing amounts of energy consumption data available to researchers in recent years make it easier to analyze the global energy demands. The analysis is important because buildings consume roughly 40% of global energy. One of the applications of data-driven analysis is identifying buildings that follow similar energy consumption patterns. Despite the number of available unsupervised machine learning algorithms, the user is still required to choose features, typically resorting to hourly data. In this paper, we propose we present SCAFE, an end-to-end architecture for Simultaneous Clustering And non-linear Feature Extraction. The method consists of first transforming the raw energy profiles that are represented as time-series into three channel images, using Gramian Summation, Gramian Difference and Markov Transition Fields. Then, we map the images to the latent space with the use of an autoencoder. Finally, we perform clustering in the latent space, simultaneously obtaining clusters and features of the dataset. We demonstrate SCAFE on the UTexas dataset, showing similar performance to standard k-means clustering while in addition also extracting natural features of the data in the latent space.

1. Introduction
Over a few recent years, the rise of the Internet of Things field made collecting of large amounts of data from buildings significantly easier with the introduction of microsensors. These amounts of data have finally allowed using the data-driven approach in the analysis of building energy consumption, which is one of the first steps of reducing buildings’ carbon footprint as they are responsible for roughly 40% of the global energy consumption and 30% of the greenhouse gas emissions [1].

Energy consumption data is best represented as time-series data as the consumption is a temporal process, capturing average energy consumption for a specific point in time. It is notoriously hard to extract features from time-series; however, some clustering methods exist that can perform the partition of the data in the time domain. In this work, we show that encoding the time-series objects like images and applying deep learning algorithms allows getting information about the individual energy profiles as well as the structure of the data. Typical clustering methods, such as k-means, perform well for partitioning a dataset. However, they do not justify the choice of clusters. Also, k-means requires to define the features a priori, and thus, does not provide any insights into the structure of the data.

On the other hand, features extracted from images using convolutional neural networks (CNN) are readily interpretable, and the process of training a CNN is stable and smooth for the most part. In this paper, we present SCAFE, an end-to-end architecture for Simultaneous Clustering And non-linear...
Feature Extraction (Figure 1). First, we transform the time-series objects into three-channel images, using the Gramian Summation (GASF) and Difference (GADF), and Markov Transition Fields (MTF). Then, we train an undercomplete convolutional autoencoder to learn the latent manifold of the underlying data. Finally, we perform the clustering, using the agglomerative clustering algorithm to partition the dataset.

Figure 1. Overview of the proposed architecture. Time series are encoded using GASF, GADF, and MTF. Then 3D tensors of encoded images form the inputs serve as targets to the Autoencoder. Features are extracted by the convolutional encoder, then transpose convolutions restore the images to learn the underlying latent manifold. Then clustering is performed on the extracted features of the dataset in latent P-dimensional space, using the Spectral non-linear clustering algorithm.

2. Methodology: SCAFE
2.1. Time-series to image transformation
We follow [2] to transform the raw time-series into images, using GASF, GADF, and MTF: One data point is represented as three matrices, one matrix from each GASF, GADF and MTF transformation. MTF encodes the temporal dependencies in the elements of its matrix. The values of the time-series are broken into the quantile bins with the desired granularity, and every point in the MTF matrix represents the transition probability from the one point to all the other points in the time domain. We then stack the three matrices along the color channel dimension to obtain three-channel images. The result of the transformation is a three-channel image for every time-series data point.

2.2. Feature extraction
SCAFE uses convolutional neural networks (CNNs) [3] to learn the features of the images. CNNs are vastly used for image processing and extensively researched. SCAFE uses a derivative architecture, the convolutional undercomplete autoencoder [4] to learn the underlying manifold of our data and extract the features of the individual time-series. An undercomplete autoencoder belongs to the family of unsupervised machine learning algorithms as it uses the same images as inputs and outputs. Undercompleteness is characterized by the existence of a bottleneck in the middle of the autoencoder, with the dimensionality of the bottleneck, typically, significantly smaller than the dimensionality of the input images. The bottleneck forces the signal to be filtered off the noise, preserving only strong correlations between the features of the data. Also, the usage of activation functions such as rectified linear unit introduces non-linearity in feature extraction.

Additionally, we use a modification of the Grad-CAM [5] algorithm to visualize some of the learned features. Features extracted during the autoencoder training non-linearly affect the choice of the resulting clustering. We use the resulting latent space vector to calculate the gradients of the vector with
respect to the activations of the latest convolutional vector. This allows us to see which features affect the most the choice of a particular partition.

2.3. Non-linear clustering
Given a trained autoencoder, the resulting latent space vector represents some non-linear combination of data features. This fact allows us to perform clustering based on the extracted features or in the latent space. We use agglomerative clustering method [6] out of the assumption that similar profiles are closer in the latent space and can be linked together with no effort.

Even though we find that agglomerative clustering performs the best in our experiments, it is possible that other clustering algorithms may perform just as well, or specific datasets require different clustering algorithms to be tried first in order to find the one that works the best for that particular dataset.

2.4. UTexas dataset
We apply SCAFE to the UTexas dataset [7] that consists of 41016 data points (a time-series object with 24 readings of energy consumption—1/hour), representing electricity metering from 129 buildings of the UT Austin campus for 2 years.

3. Results

3.1. Time-series transformation

Figure 2. One of the raw time-series and its GASF, GADF and MTF transformations.

Figure 2 shows one time-series and the three resulting matrices. Every matrix is 24 x 24 as the result of every data point being represented as readings over 24 hours. The matrices are then stacked along the channel dimension to form an image. As a result of this transformation, every energy profile in the UTexas dataset gets encoded as a three-channel image. It is possible to partially restore the original time-series object from the main diagonal of the GASF. We use this fact to project the features from the image domain back to the time domain for further feature examination.

3.2. Autoencoder
Next, we train an undercomplete convolutional autoencoder. The autoencoder consists of only four encoding layers and four decoding layers. It is highly likely that more layers may be required to cluster datasets with images of larger dimensionality such as datasets with 96 measurements per day.

We use batch normalization [8] and weight initialization [9] to improve convergence. We also use parametric rectified linear unit activation function [9] to introduce non-linearity to the autoencoder architecture. We optimize the weights of the autoencoder and other parameters with Adam [10] optimization algorithm for 300 epochs with a learning rate of 0.001 and batches consisting of 128 images each.

It is crucial to mention that the dimensionality of the latent space is an important hyperparameter of SCAFE. The resulting clusters and the quality of the extracted features may change drastically depending on the number of dimensions allowed in the latent space. It is almost certain that a different number of dimensions in the latent space may be required for different datasets, depending on the...
number of measurements taken per period. For the UTexas dataset, we found that 128-dimensional latent space works well.

3.3. Clustering
We use the Calinski-Harabasz [11] score to determine the optimal number of clusters for our dataset. This score metric represents a trade-off between separation and cohesion of clusters by using both the average between- and within-cluster sum of squares as

$$CH\text{ score} = \frac{\sum_{i=1}^{k} |C_i| ||c_i - c||^2 / (k - 1)}{\sum_{i=1}^{k} \sum_{x \in C_i} ||x - c||^2 / (n - k)}$$

where $k$ is the number of clusters, $C_i$ is cluster $i$, $x$ is a point in cluster $C_i$, $c_i$ is the centroid of cluster $C_i$, $|C_i|$ is the number of points in each cluster, $c$ is the overall centroid of the data, and $n$ is the number of data points. This metric was successfully used in a similar study [7].

Since autoencoder weights are initialized randomly along with other parameters in the architecture, we perform ten runs each for $k=2\ldots10$ of the proposed algorithm to obtain statistics on the CH score. We find that $CH(k=2) = 15900 > CH(k=j)$ for all $j=3\ldots10$. (Plot not shown for reasons of space) Thus, we use $k=2$ clusters for the UTexas dataset. Figure 3a shows the two cluster centroids as the result of our proposed method. The two profiles in Figure 3a, i.e., orange and green indicate noon-peak and night-peak profiles, respectively. As a comparison, we performed K-Means clustering with UTexas dataset using each hour as features (24 total). We obtained the best CH score (20,000) for $k=2$ as well. The two profiles from K-Means clustering are shown in Figure 3b. The results from both methods are very similar in terms of profile shape in the time domain; the green profiles are slightly different in the two results. For example, the maximum value of the green profile in Figure 3a is 0.5, whereas one in Figure 3b is 0.8.

![Figure 3. Cluster centroids for a) proposed method in this paper and b) K-Means clustering. First cluster centroid is shown in orange color and the second in green color.](image)

While the clustering results are similar, our method is superior in at least one way. Pure clustering methods such as K-Means and Expectation Maximization result in a partition over the dataset with user-defined features. However, in addition to the partition, our method also results in tractable and natural, non-linear features present in the data. These features can later be used as justification for clustering or other purposes.

3.4. Feature extraction and visualization
After training, we can visualize the features extracted from the dataset. Figure 4 shows a subset of the extracted features (activations of the third convolutional layer of the autoencoder) of one of the data
points of the UTexas dataset. Note that deeper layers of the convolutional neural network encode more abstract features as the receptive field is quite large.

Figure 4. Some of the feature maps of the third convolutional layer of the autoencoder trained on the UTexas dataset

We apply a modification of a Grad-CAM algorithm to the resulting autoencoder to obtain the heatmaps of the most significant features of the data points (See Fig.5). Instead of the scores of the classes that are used in the Grad-CAM algorithm we use the logits in the latent space, to obtain an insight of which features contribute the most to the encoding of the images.

Figure 5. Heatmaps of the feature importance for 5 data points in the UTexas dataset

The heatmaps, can be then used to transform the most important features back to the time domain, as we can partially reconstruct the original time-series from the main diagonal of the Gramian Angular Summation Field. For this, we use a threshold hyper-parameter to cut off some of the unimportant. Figure 6 shows the features obtained from the proposed algorithm with different values of threshold. Clearly, rather than hourly values as features, the algorithm chooses more continuous features inherent to the dataset.

4. Conclusion

In this paper, we introduce SCAFE an end-to-end architecture of Simultaneous Clustering And Feature Extraction. The code is available publicly from our github page [12]. We demonstrate SCAFE on the UTexas dataset, showing similar performance to standard k-means clustering while in addition also extracting natural features of the data in the latent space. Application of this algorithm alleviates the user from having to perform feature selection, which greatly simplifies the clustering procedure.

References
[1] UNEP Buildings and climate change, summary for decision-makers 2009
[2] Wang Z, and Oats T 2015 Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks. Papers from the 2015 AAAI Workshop pp 40-46
[3] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-Based Learning Applied to Document Recognition Proc. of the IEEE
[4] Masci J, Meier U, Ciresan D, and Schmidhuber J 2011 Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction
[5] Selvaraju R, Cogswell M, Das A, Vedantam R, Parikh D and Batra D 2017 Grad-CAM: Visual
Explanations from Deep Networks via Gradient-based Localization ICCV 2017

[6] Ward J 1963 Hierarchical Grouping to Optimize an Objective Function Journal of the American Statistical Association pp 236–244

[7] Park J Y, Yang X, Miller C, Arjunan P and Nagy Z 2019 Apples or oranges? Identification of fundamental load shape profiles for benchmarking buildings using a large and diverse dataset Applied Energy Volume 236 pp 1280-1295

[8] Ioffe S and Szegedy C 2015 Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

[9] He K, Zhang X, Ren S and Sun J 2015 Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification

[10] Kingma D, Ba J 2015 Adam: A Method for Stochastic Optimization ICLR

[11] T Calinski and Harabasz J 1974 A dendrite method for cluster analysis Comm in Statistics

[12] https://github.com/intelligent-environments-lab/scafe

Figure 6. The reconstructed features (orange) are first thresholded and then projected over the original time series (blue) of the corresponding data points. Different rows correspond to different levels of thresholding. The data points are the same as in Figure 5. Note that not all thresholds are suitable for all the features.