Application of Functional Feature Extraction to the Compression of Network Time Series

David Muelas, José Luis García-Dorado Jorge E. López de Vergara, Javier Aracil
High Performance Computing and Networking Research Group
Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain
Email: {dav.muelas, jl.garcia, jorge.lopez_vergara, javier.aracil}@uam.es

***Abstract*—Network management actions require the retention of data representing the temporal evolution of network state, mainly in the form of time series. Nonetheless, storing and exploiting those measurements is becoming a challenge as the production rate of such data is continuously increasing and data lasting for long time periods are used. To scale up the storage and improve both the analysis and visualization of network measurements, we apply Functional Principal Components Analysis (FPCA) to extract the most meaningful functional features for network time series, pruning those with low informational importance. We compare such algorithm with other state-of-the-art proposals, and show that it achieves lower error for the representation of atypical observations even with higher compression ratios.**

***Index Terms*—Functional Principal Component Analysis; Wavelets; Compression Algorithm**

I. Introduction

Measurements from current computer networks generate vast data volumes, many times related to time series representing network states. Even more when the volume of such network time series is expected to abruptly grow once the deployment of the Internet of Things (IoT) has reached a mature state [1]. This fact, together with management activities that consider data covering a wide time range (*e.g.*, long-term studies to detect changes in the use of infrastructures), makes the network analysis harder.

While those data provide network managers with a rich variety of information sources, their storage and manipulation are also becoming a challenge. Hence, network management actions must be supported by techniques and tools that *(i)* select pieces of information that fairly represent the network behavior, and *(ii)* reduce the data to be stored and analyzed. Actually, the importance of compression methods for thinning network time series is clearly stated in RFC 1857 [2]. Specifically, the recommendations in this document define a compaction of measurements based on the reduction of its temporal resolution —namely, 15 minutes, 1 hour and 1 day resolutions for daily, monthly and one-year observations respectively. Nonetheless, approaches relying on such methodology cannot suffice for new services and configuration capabilities that benefit from finer analysis of network dynamics. This fact motivates the exploration of strategies that reduce the volume of time series without changing the number of observations in each time interval.

As a contribution to this field, in this work we consider the transformation of network measurement time series and describe an algorithm to select their main variation modes and reduce their dimension. To do so, we rely on a functional feature extraction technique that reduces the data requirements to store time series. From a functional viewpoint we consider that time series are sampled trajectories of a continuous stochastic process —that is, punctual observations from the graph of a function $(t, F(t))$, with $F(t)$ a function in an interval of $\mathbb{R}$. Following this approach, we study those trajectories as functional random variable realizations and apply Functional Principal Components Analysis (FPCA) [3] exploiting the advantages of such methodology [4], [5].

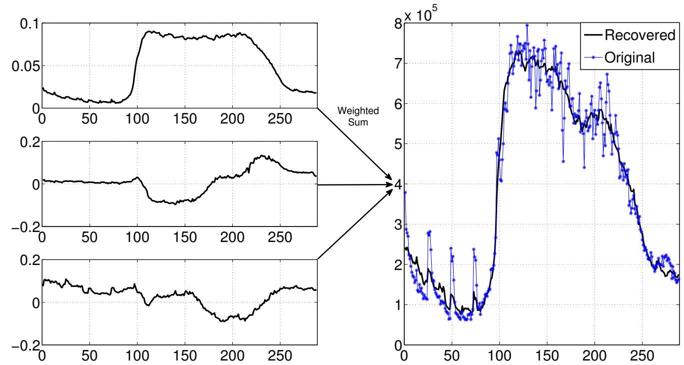


Fig. 1: Example of functional features and recovered functional observations.

Figure 1 illustrates the behavior and meaning of functional features. After extracting the functional principal components (left of the figure) the observation is recovered using a vector of weights. Hence, with this approach we decompose daily observations in terms of time series which *(i)* maximize the explained variance and *(ii)* minimize their mutual correlation. Then, we reduce the data storage requirements by retaining only a subset of the extracted principal components.

II. State of the Art

Before describing the foundations of functional feature extraction using FPCA, we review other techniques in the state of the art. For the sake of brevity, we only cover methods that have been applied to produce high dimensional compressed versions of network time series, as we focus on compression without changing the temporal resolution of data. In Section IV we thoroughly compare the accuracy of the methods here mentioned with our proposal by controlling the retained amount of data, highlighting the main advantages of FPCA.

Multiresolution analysis with wavelets [6], [7] has been used as an alternative to the thresholds recommended in RFC 1857, given the properties of such approach and its results in many other fields of applications. Wavelets are functional basis constructed using a particular oscillation (*mother wavelet*) that is scaled to obtain terms which are tied to approximations and details of the original signal. With this, the decomposition in approximations and details define an iterative transformation able to compress data by halving the data volume in each step —hereafter, we refer to the number of steps as *approximation order*. Remarkably, multiresolution analysis is a particular case of a functional consideration of time series with a fixed functional basis —instead of inferring it from observations as in the case of FPCA.

Principal Components Analysis (PCA) has been applied as a high dimensional approach to study network time series too. PCA is based in a transformation of the observations into values of a set of linearly uncorrelated variables. In [8], PCA is used to decompose throughput time series in terms of *eigenflows*, which served to analyze typical and atypical patterns in network measurements. Although they focused on anomaly detection based on such decomposition, PCA can be also used as a compression method if part of the principal components is suppressed. Although this is the same compression strategy that we follow in our method, PCA is defined in a finite dimensional vector space, while we represent the data in a functional space before the projection step.

III. Data compression algorithm

To clarify the description of our proposal, we describe its conceptual structure in Figure 2, and we summarize our notation in Table I. Roughly speaking, once a bulk of measurements is stored in the system, we represent those measurements with respect to a certain functional basis by means of interpolation —instead of using a direct projection as in the case of PCA. After this, we apply FPCA to the functional representation, and we select the most meaningful functional components (Block 1) despite of using a fixed functional basis as wavelets. Once we have that representation in terms of the principal functional components, we store the observations coefficients, the numerical evaluation of the selected functional basis, and some metadata such as the beginning and sampling times of observations (Block 2). With such information, we recover estimations of *(i)* the original observations, or *(ii)* linear transformations which are obtained by applying them to the principal components numerical evaluations (Block 3).

Algorithm 1 Data Compression

$input = \{X_t^{(j)}\},\ t \in 1,\dots,N,\ j \in 1,\dots,M$
$coefficients = \{\}$
$components = \{\}$
$P = P_0$

while $1 \leq k \leq M$ **do**
  $\hat{X}^{(k)}(t) = interpolate(X_t^{(k)}, \{B_k(t)\})$
end while

$\{PC^M(t)\} = FPCA(\{\hat{X}^{(k)}(t)\}_{k=1,\dots,M})_{k=1,\dots,M}$

if (Selection based on variance) **then**
  $P = arg(var[\{PC^P(t)\}] = var_0)$
end if

$components = evaluation(\{PC^P(t)\})$
$coefficients = projection(input\_reshape, \{PC\})$

A. Data transformation and recovery

Algorithm 1 presents the main steps of the data transformation and recovery. We consider daily time series and apply a suitable method to represent them in terms of a functional basis —*e.g.*, a Fourier basis. After that, we apply FPCA, which is described in Section III-B. This stage is purely defined in functional terms, so, once we have obtained the analytic expression of principal functional components, we evaluate them in the interval of definition. The number of functional principal components can be selected by *(i)* taking into account the explained variance or *(ii)* using limits related to retained data ratios —as we will explain in Section IV-A. To obtain the coefficients for each observation, we apply a projection of the original data in terms of the functional principal components evaluation.

TABLE I: Notation

Symbol	Definition
H	Data transformation.
T	Time that observations last.
N	Number of terms in a whole observation.
M	Number of observations.
P	Number of functional principal components.
$\{B_k(t)\}_{k\in\mathbb{Z}}$	Functional basis (infinite dimension).
$X_t^{(i)}$	Discrete time observation (time series).
$\hat{X}^{(i)}(t)$	Continuous time observation (functional).
$\{PC^P(t)\}$	Set of P principal components.
$R_t^{(i)}$	Recovered time series.

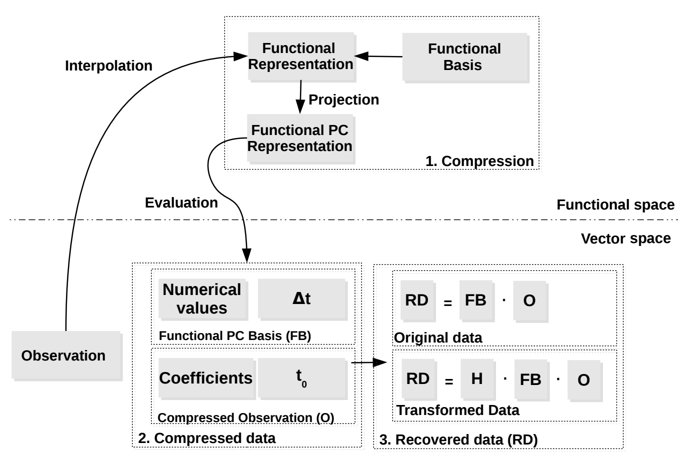


Fig. 2: Conceptual structure of our algorithm.

Interestingly, the original data can be non-uniformly sampled, which makes this approach flexible for observations that are not equally distributed over time. Additionally, if we consider linear transformations of data by using any operator H, it is enough to transform only the elements in the functional basis instead of all the observations. Furthermore, as differential transformations can be locally approximated with suitable linear transformations, they can be also optimized if some additional error is acceptable.

B. FPCA definition

The first step of our algorithm is to interpolate measurements using a certain functional basis: hereafter, we represent it with the notation $\{B_k(t)\}_{t\in\mathbb{T},k\in\mathbb{Z}}$, with $\mathbb{T}$ a real interval; and the projections obtained from observations as $\{\beta_k\}_{k\in\mathbb{Z}}$. The number of elements in the basis is truncated to get a computationally affordable representation: as a result, the considered functional representation follows Eq. 1, with $\mathbb{J}$ the particular finite index set and ϵ an error term which is dependent on both the selected index set and the specific functional basis.

$$\hat{X}(t) = [\sum_{j\in\mathbb{J}} \beta_j B_j(t)] + \epsilon(\mathbb{J}, \{B_j\}),\ t \in \mathbb{T}, \tag{1}$$

We then apply FPCA [3] to compress data using that functional representation. As in the multivariate case, FPCA is performed by projecting the original basis on a different space to maximize the explained variance while minimizing the correlation between the components. We recall that in the FDA context, we have functions $X_i(t)$ instead of multivariate variable values. Then, in the FPCA definition, the discrete index of each dimension of the multivariate case is changed by a "continuous index" t. Thus, the weights of the transformation $\{\xi_j(t)\}$ are functions in L^2. In this setup, the scores corresponding to each principal component are given by Eq. 2.

$$f_i = \int \xi(s)\hat{X}^{(i)}(s)ds = \int \xi\hat{X}^{(i)} \tag{2}$$

The weight functions $\xi_k(s)$ are chosen to satisfy Eq. 3, where we assume data $\hat{X}^{(1)}(t), \dots, \hat{X}^{(M)}(t)$ are centered.

$$\begin{cases} \frac{\sum_i f_{i_1}^2}{M} = \frac{\sum_i \int(\xi_1\hat{X}^{(i)})^2}{M} \\ \int \xi_1^2 = 1, \int \xi_k\xi_m = 0,\ \forall\ k < m. \end{cases} \tag{3}$$

IV. Evaluation

We have led a comparative between our algorithm, PCA and wavelets. For all of them, we have used the available MATLAB implementations[1], and for illustrative purposes, our code is available under request. We have used a Fourier basis at the first interpolation step for FPCA, and Daubechies 32 filters for wavelets —as these combinations provided the best results in a previous evaluation.

We have applied these methods to the compression of daily observations sets, ranging from 30 to 240 days. These sets have been constructed using time series randomly selected from a bulk of measurements from a concrete backbone router of the Spanish National Research and Education Network (RedIRIS) sampled every 5 minutes. To attenuate the variance of the results depending on the selected days, we have aggregated the outputs from 20 repetitions of the tests.

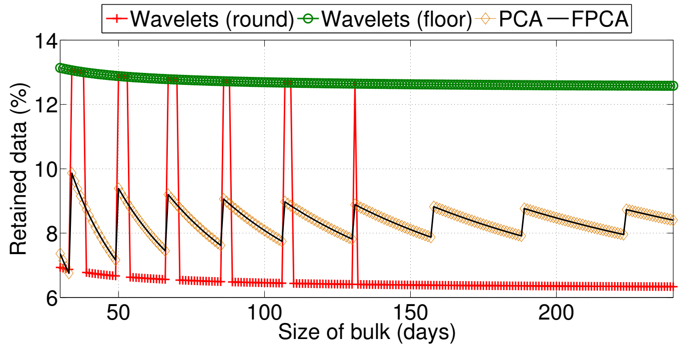


Fig. 3: Percentage of retained data for each method.

A. Data compression

The data that our algorithm retains (RD_{FPCA}) is measured as the relation between the compressed and original number of terms. It is explicitly obtained with the expression in Eq. 4, when N numerical values of the principal components are stored per observation —we use the notation in Table I.

$$RD_{FPCA} = \frac{P\cdot M + P\cdot N}{N\cdot M} = \frac{P\cdot(M+N)}{N\cdot M} \tag{4}$$

Additionally, we can evaluate the asymptotic value of the retained data ratio ($RD_{FPCA}(m)$) with the expression in Eq. 5. As a result, we obtain a bound for the minimum theoretical retained data ratio when $m \to \infty$:

$$RD_{FPCA}(m) = \frac{P\cdot(m+N)}{N\cdot m} \to \frac{P}{N} \tag{5}$$

If we compare this ratio with those of the methods that we mentioned above, we realize that PCA provides the same values, whereas the achievable data reduction using wavelets cannot be easily adjusted to them. To do so, we consider the approximation order to be used, following the relation in Eq. 6.

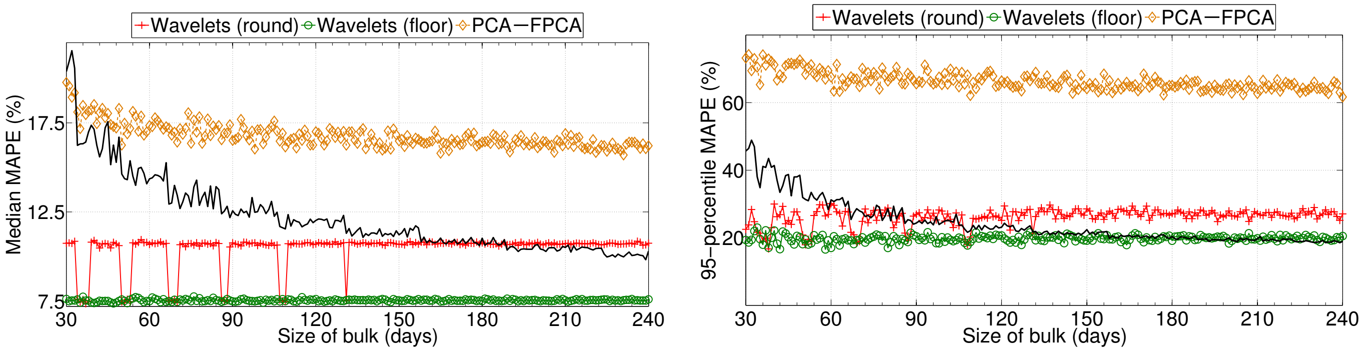
$$O_{approximation} = \log_2\left(\frac{M\cdot N}{P\cdot(M+N)}\right) \tag{6}$$

We convert $O_{approximation}$ to an integer by rounding or truncating, and test both situations in our comparative. To compare the data reduction in each case, we show the actual ratio that we obtain for each size and method in Figure 3. Remarkably, the ratio of retained data of our algorithm and PCA is a 2-3% higher and a 3-7% lower (with respect to the original data size) than those corresponding to wavelets after a rounded or truncated number of approximations, respectively.

B. Accuracy comparison

In the following, we analyze the information loss for each method using the dissimilarity between the reconstructed and

[1]http://www.psych.mcgill.ca/misc/fda/downloads/FDAfuns/Matlab/



(a) Median MAPE

(b) 95-percentile MAPE

Fig. 4: MAPE for each method when varying the size in days of the data bulk.

the original time series. Our evaluation of dissimilarity is based on the Mean Absolute Percentage Error (MAPE), which is an accuracy metric given by the expression in Eq. 7. Specifically, this metric characterizes the global behavior of the recovered data by estimating the expected punctual relative errors with respect to the original data.

$$M_i = \sum_{j=1}^{N} \frac{|X_{t_j}^{(i)} - R_{t_j}^{(i)}|}{X_{t_j}^{(i)}} \cdot \frac{100}{N} \qquad (7)$$

Figure 4 presents the values of the median and 95-percentile MAPE —the selection of those levels is justified, as the estimation improves when the MAPE values are low. Additionally, with these levels we can characterize the performance of each algorithm both for daily observations with typical and atypical patterns. The results show that our approach is stable for bulks with more than 180 days, as the definition of the principal components requires collecting enough data to obtain significant results and the included number of principal components increases with the size of the bulk. Then:

- FPCA represents typical observations better than wavelets using a rounded number of approximations, which improves the compromise between compression and information loss, and offers more flexibility for the selection of the compression and error levels.
- It also improves the representation of observations showing atypical errors, even when compared to wavelets with lower compression. Taking into account Figure 3, this means that our method stores a 5-6% less of the original volume and, as a result, it reduces the required storage around 1.5 times with respect to the latter.

V. Conclusions and Future Work Lines

We have presented an algorithm that exploits FPCA as a compression technique that improves long-term forensic studies using network time series and makes easier the manipulation and transformation of data using linear operators. We have compared its performance in terms of data volume reduction and accuracy with other methods previously applied in the network time series analysis scope. Our approach achieves better results for bulks lasting more than 180 days than the wavelet solution with the most similar ratio, and improves the representation of observations with the worst dissimilarity levels, even with lower retained data ratios.

Regarding future work lines, the quantitative and qualitative evaluation of traffic patterns' impact stands out as a promising field that may open the gate to better compression approaches. Moreover, we plan to expand our analysis to the reduction of live data, by applying time series' segmentation strategies and with the exploitation of adaptive algorithms based on the methods which are compared in this work.

Acknowledgment

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund under the project TRÁFICA (MINECO/FEDER TEC2015-69417-C2-1-R).

References

[1] A. Papageorgiou, B. Cheng, and E. Kovacs, "Reconstructability-aware filtering and forwarding of time series data in internet-of-things architectures," in *Proc. IEEE International Congress on Big Data (BigData 2015)*, June 2015, pp. 576–583.

[2] M. H. Lambert, "A Model for Common Operational Statistics," IETF Request For Comments 1857, Oct. 1995.

[3] J. Ramsay and B. Silverman, *Functional Data Analysis*. Springer, New York, 1997.

[4] D. Muelas, J. E. López de Vergara, and J. R. Berrendero, "Functional data analysis: A step forward in network management," in *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*, May 2015, pp. 882–885.

[5] D. Muelas, J. E. López de Vergara, J. R. Berrendero, J. Ramos, and J. Aracil, "Facing network management challenges with functional data analysis: Techniques & opportunities," *Mobile Networks and Applications*, pp. 1–13, 2016.

[6] J. L. García-Dorado, J. Aracil, J. A. Hernández, and J. E. López de Vergara, "A queueing equivalent thresholding method for thinning traffic captures," in *Proc. IEEE Network Operations and Management Symposium (NOMS 2008)*, April 2008, pp. 176–183.

[7] K. Kyriakopoulos and D. Parish, "A live system for wavelet compression of high speed computer network measurements," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4427, pp. 241–244.

[8] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 61–72, Jun. 2004.