The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Non-Parametric Outliers Detection in Multiple Time Series A Case Study: Power Grid Data Analysis

Yuxun Zhou,[*] Han Zou,[*] Reza Arghandeh,[†] Weixi Gu,[‡] Costas J. Spanos[*]
[*]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA
[†]Department of Electrical and Computer Engineering, Florida State University, USA
[‡] Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, China
Email: {yxzhou, hanzou, spanos}@berkeley.edu, arghandehr@gmail.com, guweixigavin@gmail.com

Abstract

In this study we consider the problem of outlier detection with multiple co-evolving time series data. To capture both the temporal dependence and the inter-series relatedness, a multi-task non-parametric model is proposed, which can be extended to data with a broader exponential family distribution by adopting the notion of Bregman divergence. Albeit convex, the learning problem can be hard as the time series accumulate. In this regards, an efficient randomized block coordinate descent (RBCD) algorithm is proposed. The model and the algorithm is tested with a real-world application, involving outlier detection and event analysis in power distribution networks with high resolution multi-stream measurements. It is shown that the incorporation of inter-series relatedness enables the detection of system level events which would otherwise be unobservable with traditional methods.

1 Introduction

Data sets collected from a wide variety of research disciplines, including computer science, economic, biology and social science, are in the form of multiple co-evolving time series. In this work, we consider the task of outlier (or novelty) detection given the aforementioned data type. The core difficulty, however, is to integrate both the temporal dependence and the interactions among correlated time series for overall modeling and learning.

General outlier detection is a broad topic that is usually studied separately in the context of particular domain application. From a statistical learning perspective, however, outlier detection techniques can be categorized according to their input data types, including but not limited to independent and identically distributed observations (Aggarwal and Yu 2008; Zhou et al. 2016), high-dimensional data (Aggarwal and Yu 2001), time series (Gupta et al. 2014), structural data such as graphs and network (Aggarwal, Zhao, and Philip 2011; Gupta et al. 2012), etc. A detailed exposition of general outlier detection techniques is beyond the scope of this paper. The readers are referred to (Aggarwal 2015) and the references therein for an extensive overview. Depending on different views of the data generating process, methods for outlier detection in time series can be summarized into the following categories:

Physical model based methods. The underlying assumption is that the observed time series data is generated from a known dynamic system. As such, the problem is reduced to comparing the system behavior, estimated with measurements and dynamic equations, to the expected behavior when the system is in a certain state (normal or abnormal)(Isermann 2005). Recently, model based approaches have also been used in combination with time series analysis to establish semi-model based algorithms (Cavraro et al. 2015a; 2015b). This type of approaches rely heavily on correctness of the dynamical model of the system, as well as some system analytic tools such as real-time state estimators, parameter estimation, parity equations, etc. Their limitations are obvious particularly as recent applications have to deal with high dimensional and inherently uncertain processes, which significantly deteriorates the reliability and accuracy of dynamic models.

Signal processing based filtering methods. Those approaches implicitly assume that the "normal" component of the time series has a sparse representation in the frequency or wavelet domain. Hence the outlier detection problem is reduced to a spectral analysis using low pass or band pass filters, or is solved by denoising/signal reconstruction using spectral or wavelet techniques (Mallat 2008). It is worth pointing out that the signal-processing-based methods have close ties with the regularized basis function expansion method in statistical learning. For example, the adaptive wavelet denoising method known as SURE shrinkage (Donoho and Johnstone 1995) is essentially the L_1 regularized wavelet basis expansion.

Statistical learning based method. The key is to model the characteristics of the normal state, e.g., the support of its distribution, its sparse representation, or its smooth component, with certain parametric or non-parametric learning tools. As a large amount of data is made available by the advancements in sensing and measurement technology, this approach is receiving increasing attention in both application and research domains. Ignoring the temporal dependence, many classic machine learning tools, such as the Kernel Principle Component Analysis (kPCA), one class SVM, etc., have been widely applied to various fields for outlier detection. When the temporal dependece is informative (Zhou and Spanos 2016), miscellaneous time series modeling and analysis tools, ranging from simple linear regression to complicated multivariate AMRIA models and from parametric dynamic Bayeisan networks to non-parametric regression methods, can be adopted. Readers are referred to (Aggarwal 2015) and the references therein for a comprehensive survey.

However, few works have addressed the outlier detection problem for multiple correlated time series. In this work, we propose a non-parametric learning framework, by extending the classical smoothness (complexity) and fitness optimization. The relatedness among series is captured with an additional regularization term that imposes the smoothness of "aggregated pair-wise difference". We also show that the framework can be readily extended to time series with miscellaneous types, with the introduction of exponential family and Bregman divergence. Moreover, an efficient randomized block coordinate descent (RBCD) algorithm is proposed and analyzed to alleviate the computational difficulty of the non-parametric model learning for large data set. To complement the theoretical analysis, the non-parametric model and the RBCD algorithm is implemented in a real-world application, involving outlier detection and event analysis in power distribution networks. The high resolution (millisecond) and high dimensionality of the multi-stream measurements obtained in this application posses a challenging outlier detection problem. It is shown that the proposed framework, which incorporates the inter-series relatedness, enables the detection of system level events which would otherwise be unobservable with traditional methods. Moreover, the proposed RBCD algorithm scales much better in computational cost compared to other alternatives.

The rest of the paper is organized as follows. The next section is devoted to formulating a non-parametric model of multiple time series. Also, an extension of the model to exponential family is elaborated to deal with time series with miscellaneous distributions. In section 3, the RBCD algorithm is established and analyzed to learn the non-parametric model from data. Finally, we describe the application context and demonstrate the performance of the proposed method.

2 A Non-parametric Model for Multiple Time Series

2.1 Notation and Definition

Before proceeding to any technical details, we standardize our notation by using a matrix $X^{M\times T}$ to represent multiple time series measurements for T time steps and M streams. Note that for sensor network applications we usually have $M = K \times L$ where K is the number of channels of each sensor and L the number of sensors installed in the network. To represent the dependence among streams, a "contextual" matrix $C^{M\times M}$ is designated to store the pair-wise correlations. Also for the ease of discussion, we adopt the notion of Network of Time Series (NoT):

Definition 1. *A Network of Time Series (NoT) is defined as the triplet $\mathcal{G} = \{X, C, d\}$, where $X \in \mathbb{R}^{M\times T}$ is a collection of M time series of T time steps, $C \in \mathbb{R}^{M\times M}$ is the contextual matrix and d a dictionary that maps each dimension or stream of X to an entry in C.*

As far as outlier detection is concerned, we adopt the convention that the abnormality or novelty of an observation X_{it} is defined as the deviation between estimated (expected) value $\widehat{X}_{it}$ and real measurement X_{it}. Hence the problem of novelty detection reads,

Problem 1. *Given $\mathcal{G} = \{X, C, d\}$, estimate $\widehat{X}_{it}$, $\forall i, t$. Then compute $l\left(X_{it}, \widehat{X}_{it}\right)$ as the index of novelty, where $l(\cdot,\cdot)$ is a metric function $\mathbb{R}\times\mathbb{R}\mapsto\mathbb{R}$.*

Hence the core of the novelty detection problem is an estimation problem, for which both temporal dependence and inter-series correlation should be taken into account.

2.2 Intuition and Problem Formulation

The method we propose borrows ideas from two separate yet closely related research domains, i.e., time series de-trending in economics and non-parametric regression in statistical learning. The learning formulation, proposed in this part for multiple time series data, is quite intuitive and can be extended to other data types with the introduction of the Bregman Divergence. We start by considering the following decomposition for a single time series x_t:

$$x_t = u_t + w_t \qquad \forall t \tag{1}$$

where the new time series u_t represents the trend component in the terminology of economics, and the second term w_t contains the so called cyclical component and noises of the original time series (Enders 2004)[1]. As such, outlier or novelty can be defined as elements that deviate significantly from the general trend. In order to find the trend component, one can simply optimize over a "fitness" and "smoothness" trade-off:

$$\min_{u_0,\cdots,u_T} \sum_{t=1}^{T} l(x_t, u_t) + \lambda\Omega(u_0,\cdots,u_T) \tag{2}$$

where $l(\cdot,\cdot)$ and $\Omega(\cdot)$ are loss functions imposed on "fitness" and "smoothness", respectively. The above formulation is also closely related with the non-parametric regression method in statistical learning (Fan and Gijbels 1996), in which a regression function is found by minimizing the L_2 loss with second-order derivative regularization. Similarly, when dealing with time series data containing discrete-time, continuous-value records, one can substantiate the objective (2) as follows:

$$\min_{u_0,\cdots,u_T} \sum_{t=1}^{T}(x_t - u_t)^2 + \lambda\sum_{t=1}^{T}(\nabla_t^2 u_t)^2 \tag{3}$$

where ∇_t^2 is the second order difference operator defined by:

$$\nabla_t^2 u_t = \begin{cases} 0 & t = 1 \\ u_{t+1} + u_{t-1} - 2u_t & 2 \le t \le T-1 \\ 0 & t = T \end{cases} \tag{4}$$

Like the second order derivative regularization used in non-parametric regression, the above aggregated second order

[1] Hence one can decompose this term into $w_t = c_t + \varepsilon_t$ for further analysis

differences also measures the smoothness of the entire sequence, hence is sometimes referred to as total variation regularization. Detailed analysis and more statistical property of this term can be found in (Harchaoui and Lévy-Leduc 2010).

By solving the convex quadratic optimization problem (3), one is able to find the trend component u_t. Any data point that significantly deviate from the trend is an outlier or novelty point. The weighting parameter λ is called the smoothness parameter, which should be tuned according to the application purpose using model selection techniques. It is worth pointing out that the solution to (3) is called the Hodrick-Prescott filter in economic time series analysis (Hodrick and Prescott 1997).

Now we establish our model by extending the above nonparametric framework to multiple time series that are correlated with each other. Notation-wise, given multiple time series data $X \in \mathbb{R}^{M \times T}$, we denote the t^{th} element of the m^{th} time series by x_{mt}, i.e., x_{mt} is the $(m, t)^{th}$ entry of the data matrix X. Also, the boldface $\boldsymbol{x}_m$ is used to represent the row vector $[x_{m1}, \cdots, x_{mT}]$. Similarly, $\boldsymbol{u}_m = [u_{m1}, \cdots, u_{mT}]$. Consider minimizing the following objective:

$$\min_{\boldsymbol{u}_1,\cdots,\boldsymbol{u}_M} \sum_{m=1}^{M}\sum_{t=1}^{T}(x_{mt}-u_{mt})^2 + \lambda_1 \sum_{m=1}^{M}\sum_{t=2}^{T-2}\left(\nabla_t^2 u_{mt}\right)^2 + \lambda_2 \sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M}\sum_{t=2}^{T-1}\left[\nabla_t^2(u_{it}-C_{ij}u_{jt})\right]^2 \tag{5}$$

where λ_1 and λ_2 are two regularization hyper-parameters, and C is the standardized co-variance matrix with entries

$$C_{ij} = \mathrm{cov}(\boldsymbol{x}_i, \boldsymbol{x}_j)\,(\mathrm{var}(\boldsymbol{x}_j))^{-1} \tag{6}$$

The intuition for the first two terms in (5) is straightforward: we simply aggregate the fitness and smoothness objectives of M times sequences. The motivation for the third term is the following: Since the linear least square estimator (LLSE) (Chatterjee and Hadi 2015) of $\boldsymbol{u}_i$ given $\boldsymbol{u}_j$ reads

$$\mathbb{E}[\boldsymbol{u}_i] - \mathrm{cov}(\boldsymbol{u}_i, \boldsymbol{u}_j)\,(\mathrm{var}(\boldsymbol{u}_j))^{-1}\,(\boldsymbol{u}_j - \mathbb{E}[\boldsymbol{u}_j]).$$

In the case where the two trends are ideally correlated, $\boldsymbol{u}_i - \mathrm{cov}(\boldsymbol{u}_i, \boldsymbol{u}_j)\,(\mathrm{var}(\boldsymbol{u}_j))^{-1}\,\boldsymbol{u}_j$ should be a constant sequence. Consider estimating the covariance of U by that of the noisy X, and relax the harsh "constant" requirement to smoothness, then with the same usage of second order difference, the third term imposes the smoothness of the sequence $\boldsymbol{u}_i - C_{ij}\boldsymbol{u}_j$, which is aggregated over all pairwise combinations.

2.3 Extension to Exponential Family

The previous discussion is focused on time series having continuous values. Many time sequences, however, may contain non-negative or categorical values depending on the practical data generating process. Given that consideration, we extend the smoothing method to time series with exponential family marginal distributions. It is helpful to recall some definitions to begin with:

Definition 2. *The **Bregman Divergence** of any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, with respect to some arbitrary differentiable strictly convex function $F : \mathbb{R}^n \to \mathbb{R}$ is defined by*

$$B_F(\boldsymbol{x}, \boldsymbol{y}) = F(\boldsymbol{x}) - F(\boldsymbol{y}) - (\boldsymbol{x} - \boldsymbol{y}) \cdot F'(\boldsymbol{y}) \tag{7}$$

One can think of the Bregman Divergence as simply the nonlinear tail of the Taylor expansion of $F(\boldsymbol{x})$ around $\boldsymbol{y}$. Note that the Bregman Divergence is not symmetric, however, it holds that $B_F(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$.

Definition 3. *A family of distributions is said to belong to **Exponential Family** in canonical form if the probability density function, or probability mass function for discrete distributions, can be written as*

$$f_X(\boldsymbol{x}|\boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\{\boldsymbol{\theta} \cdot T(\boldsymbol{x}) - A(\boldsymbol{\theta})\} \tag{8}$$

where the parameter vector $\boldsymbol{\theta}$ is called the natural parameter of the distribution, and $T(\boldsymbol{x})$ the sufficient statistic. We also denote $a(\boldsymbol{\theta}) \triangleq A'(\boldsymbol{\theta})$ for future use.

When Bregman Divergence is used in measuring the fitness of observed data to a parametrized exponential family distribution, the following property shows that Bregman divergence is directly related with log-likelihood:

Theorem 1. *Define a dual function associated with the exponential family*

$$F(a(\boldsymbol{\theta})) \triangleq \boldsymbol{\theta} \cdot a(\boldsymbol{\theta}) - A(\boldsymbol{\theta}) \tag{9}$$

then $F(\mu)$ is strictly convex in $\mu = a(\boldsymbol{\theta})$. In addition,

$$B_F(T(\boldsymbol{x})||a(\boldsymbol{\theta})) \propto -logP(T(\boldsymbol{x})|\boldsymbol{\theta}) \propto A(\boldsymbol{\theta}) - T(\boldsymbol{x}) \cdot \boldsymbol{\theta} \tag{10}$$

Proof. Treating $\boldsymbol{\theta}$ as a function of μ, and taking derivative of $F(a(\boldsymbol{\theta}))$ with respect to μ, we get

$$\nabla_\mu F(\mu) = f(\mu) = \boldsymbol{\theta} + \frac{\partial \boldsymbol{\theta}}{\partial \mu}\mu - \frac{\partial \boldsymbol{\theta}}{\partial \mu}\mu = \boldsymbol{\theta} \tag{11}$$

which is in effect the inverse of $a(\boldsymbol{\theta})$, i.e.,$\nabla_\mu F(\mu) = \boldsymbol{\theta} = a^{-1}(\mu)$. From the strict convexity of $A(\boldsymbol{\theta})$, it is guaranteed that this inverse always exists. Moreover, since $a(\boldsymbol{\theta})$ has a positive definite Jacobian, its inverse $a^{-1}(\mu)$ also has a positive definite jacobian. Hence $F(\mu)$ is strictly convex. In real analysis, $F(\mu)$ is also called the dual convex function of $A(\boldsymbol{\theta})$. Using this function in the Bregman divergence for $\boldsymbol{x}$ and $a(\boldsymbol{\theta})$, we get

$$\begin{aligned} &B_F(T(\boldsymbol{x})||a(\boldsymbol{\theta})) \\ &= F(T(\boldsymbol{x})) - F((\boldsymbol{\theta})) - (T(\boldsymbol{x}) - a(\boldsymbol{\theta})) \cdot \nabla F(a(\boldsymbol{\theta})) \\ &= F(T(\boldsymbol{x})) - a(\boldsymbol{\theta}) + A(\boldsymbol{\theta}) - (T(\boldsymbol{x}) - a(\boldsymbol{\theta})) \cdot \boldsymbol{\theta} \\ &= F(T(\boldsymbol{x})) + A(\boldsymbol{\theta}) - T(\boldsymbol{x}) \cdot \boldsymbol{\theta} \end{aligned}$$

On the other hand, since the log likelihood of the exponential family is just

$$\log P(T(\boldsymbol{x})|\boldsymbol{\theta}) = \log h(\boldsymbol{x}) + T(\boldsymbol{x}) \cdot \boldsymbol{\theta} - A(\boldsymbol{\theta})$$

Hence we can directly relate negative log likelihood and Bregman divergence by

$$B_F(T(\boldsymbol{x})||a(\boldsymbol{\theta})) = -\log P(T(\boldsymbol{x})|\boldsymbol{\theta}) + \log h(\boldsymbol{x}) + F(T(\boldsymbol{x}))$$

□

Now consider arbitrary time series $\{x_{1m}, x_{2m}, ..., x_{Tm}\}$ in the data set, whose marginal distribution (for each x_{mt}) belongs to some exponential family, a natural extension of the "fitness" loss is the Bregman divergence. Together with the above discussion, the first term in the proposed multiple time series smoothing formulation (5) could be generalized as follows

$$\begin{aligned} l(\Theta) &= \sum_{m=1}^{M}\sum_{t=1}^{T} B_F\left(T(x_{mt})||a(\theta_{mt})\right) \\ &\propto \sum_{m=1}^{M}\sum_{t=1}^{T} -\log P(T(x_{mt})|\theta_{mt}) \qquad (12) \\ &\propto \sum_{m=1}^{M}\sum_{t=1}^{T} \{A(\theta_{mt}) - T(x_{mt})\theta_{mt}\} \end{aligned}$$

where we use the matrix $\Theta \in \mathbb{R}^{M \times T}$ to denote all natural parameters associated with the elements of the multiple times series. Since natural parameters uniquely characterize the exponential family distribution, in particular its moments through cumulant function, it appears reasonable to adopt a similar regularization as in (5) for natural parameters of each entry, to impose temporal smoothness on each time sequence, as well as their inter-correlations. As such, the overall learning objective of general multiple time series smoothing reads

$$\begin{aligned} \min_{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M} \mathcal{J}(\Theta) &= \sum_{m=1}^{M}\sum_{t=1}^{T} \{A(\theta_{mt}) - T(x_{mt})\theta_{mt}\} \\ &+ \lambda_1 \sum_{m=1}^{M}\sum_{t=2}^{T-2} \left(\nabla_t^2 \theta_{mt}\right)^2 \qquad (13) \\ &+ \lambda_2 \sum_{i=1}^{M}\sum_{j=1, j \neq i}^{M}\sum_{t=2}^{T-1} \left[\nabla_t^2(\theta_{it} - C_{ij}\theta_{jt})\right]^2 \end{aligned}$$

which is still convex since the second order derivative of each component of the first term is $a'(\theta_{mt}) = \text{Var}(T(x_{mt})) > 0$.

3 A Fast Random Block Coordinate Descent (RBCD) Algorithm

So far the problem of multiple time series smoothing has been reduced to solving a convex optimization problem (13) with smoothness penalty λ_1 and λ_2 as hyper-parameters. Generic methods, such as those based on first or second order gradient (Boyd and Vandenberghe 2004; Bertsekas et al. 2003), may be applied but may not be a good choice - the dimension of the decision variables Θ equals to the number of elements of all time series, hence the calculation or even the storage of full first/second order gradient is quite inefficient. Moreover, batch gradient methods suffers from the choice of step size and numerical instability when dealing with high-dimensional problems.

In this section, we propose a simple yet efficient algorithm that can be implemented in just a few lines of code. The key idea is the archetype of an universal solution methodology to algorithmic optimization: solving a complex or large scale problem by reducing it to a sequence of simpler optimization problems. More specifically for (13), it appears that fixing all the other decision variables except $\boldsymbol{\theta}_t$, (which are the decision variables corresponding to all observations of the multiple time series at time t), the sub-problem has low dimension and the solution can be updated easily with much less time and memory. We provide a convergence analysis of the proposed RBCD algorithm, and demonstrate its relation to stochastic gradient descent (SGD). In addition, RBCD is readily amendable for parallel computation, and empirically outperforms the state-of-the-art alternating direction method of multipliers (ADMM) that was recently proposed for total variation regularized problems (Boyd et al. 2011; Gonçalves, Von Zuben, and Banerjee 2016; Zhou, Kang, and Spanos 2017).

The RBCD start with an initial guess of the decision variables Θ^0. In each step, it consists of (1) picking up an index i_k from $\{1, \cdots, T\}$, (2) evaluating the gradient of a block of variables, i.e., $[\nabla\mathcal{J}(\Theta)]_{i_k}$ in the current implementation, followed by (3) updating the i_k^{th} column of Θ. Note that we have adopted the "subset indexing" convention: here and throughout, $[\nabla\mathcal{J}(\Theta)]_i$ is used to denote the i^{th} column of $\nabla\mathcal{J}(\Theta)$. The indicator vector v_i has dimension $T \times 1$ and all its elements, except the i^{th} entry, equal to zero. The multiplication with v_i^T serves to match the dimension of block gradient to the dimension of all decision variables. Also it is worth pointing out that in each step i_k could be chosen randomly, as in the current implementation, for the purpose of parallel computing. Alternatively i_k can be selected in a deterministic fashion, e.g., using a cyclic schedule. The convergence analysis in later part of this section holds for both cases.

Algorithm 1 Random Block Coordinate Descent (RBCD) Algorithm

```
Initialize $\Theta^0 = [\boldsymbol{\theta}_1^0, ..., \boldsymbol{\theta}_T^0] \in \mathbb{R}^{M \times T}$, and let $k \leftarrow 0$
while $k < iter_{max}$ do
    Sample $i_k \in \{1, \cdots, T\}$ from a uniform distribution
    $\Theta^{k+1} \leftarrow \Theta^k + \alpha_k [\nabla\mathcal{J}(\Theta)]_{i_k} v_{i_k}^T$
    if $||\Theta^{k+1} - \Theta^{k-T+2}|| < threshold$ then
        Return $\Theta$
    end if
    $k \leftarrow k + 1$
end while
```

Now we calculate the gradients that are required by the algorithm. To begin with, the three terms of the objective function (13) are denoted by $l(\Theta)$, $\Omega_1(\Theta)$ and $\Omega_1(\Theta)$, respectively, i.e., the objective function is rewritten as

$$\mathcal{J}(\Theta) = l(\Theta) + \lambda_1\Omega_1(\Theta) + \lambda_2\Omega_1(\Theta) \qquad (14)$$

for clarity. When all elements of Θ except the i^{th} column $\boldsymbol{\theta}_i$ are fixed, we can easily compute

$$\frac{\partial l(\Theta)}{\partial \boldsymbol{\theta}_i} = -\left(a(\boldsymbol{\theta}_i) - T(\boldsymbol{x}_i)\right) \qquad (15)$$

where the function operation should be interpreted component-wise, i.e.,

$$a(\boldsymbol{\theta}_i) \triangleq [a(\theta_{1i}), \cdots, a(\theta_{Mi})]^T$$

The gradient computation of the second term is also straightforward,

$$\frac{\partial \Omega_1(\Theta)}{\partial \boldsymbol{\theta}_i} = \phi(B)\boldsymbol{\theta}_i \tag{16}$$

where $\phi(B) = B^2 - 4B + 6 - 4B^{-1} + B^{-2}$ and B is the time delay operator. The gradient of the third term is more involved, with some algebra we get

$$\frac{\partial \Omega_2(\Theta)}{\partial \boldsymbol{\theta}_i} = \phi(B)\boldsymbol{\theta}_i \cdot \left[(M-3)I + 2C + \text{diag}\left(\sum_{j=1}^{M} C_{1j}^2, \cdots, \sum_{j=1}^{M} C_{Mj}^2\right)\right] \tag{17}$$

Now we provide the convergence analysis of the algorithm.

Theorem 2. *The gradient function $\nabla \mathcal{J}(\Theta)$ is block-wise Lipschitz continuous. Let L_i be the Lipschitz constant of block i, then*

$$\begin{aligned} L_i &\geq (2 + 12\lambda_2 + 2\lambda_2(M-3)) + 2\|C\|_2 + \\ &\quad \min\{\sum_{j=1}^{M} C_{1j}^2, \cdots, \sum_{j=1}^{M} C_{Mj}^2\} \triangleq \bar{L}_{min} \quad \forall i \\ L_i &\leq (2 + 12\lambda_2 + 2\lambda_2(M-3)) + 2\|C\|_{\mathcal{F}} + \\ &\quad \max\{\sum_{j=1}^{M} C_{1j}^2, \cdots, \sum_{j=1}^{M} C_{Mj}^2\} \triangleq \bar{L}_{max} \quad \forall i \end{aligned} \tag{18}$$

The RBCD algorithm with constant step size $\alpha_k = \bar{L}$ generates a sequence $\{\Theta^k\}_{k\geq 0}$ that achieves

$$\mathbb{E}[\mathcal{J}(\Theta^k)] - \mathcal{J}^* \leq \left(1 - \frac{\bar{L}_{min}}{T\bar{L}_{max}}\right)^k (\mathcal{J}(\Theta^0) - \mathcal{J}^*) \tag{19}$$

The proof is long and technical hence is saved to the supplementary. Interestingly, the proposed RBCD method is closely related to the Stochastic Gradient Descent (SGD) method which has received much attention for large scale machine learning application. SGD tries to minimize a smooth function f by taking a negative step along an estimate g of the gradient $\nabla f(x)$. Under regular conventions, it is assumed that g is unbiased, i.e., $\mathbb{E}[g] = \nabla f(x)$, where the expectation is taken over the random variables that are used to obtain g at current value of x. The proposed RBCD method, somewhat surprisingly, can be viewed as a special case of the above SGD. In fact, if we take $g = T[\nabla \mathcal{J}(\Theta)]_{i_k} v_{i_k}^T$, then with the random sampling of the coordinate index, we have

$$\mathbb{E}[g] = \frac{1}{T}\sum_{i=1}^{T} T[\nabla \mathcal{J}(\Theta)]_i v_i^T = \nabla \mathcal{J}(\Theta) \tag{20}$$

The difference is that RBCD algorithm can guarantee an improvement of the objective function at each step. However, both the SGD and the RBCD proposed here avoid the process the entire data set at each step, hence they are both scalable for large size problems. Next we test the proposed model and algorithm in a real-world application, demonstrating the benefit of including inter-series relatedness and the effectiveness of the RBCD algorithm.

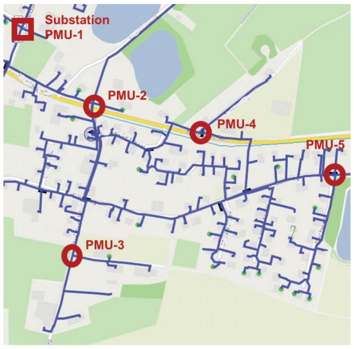

Figure 1: Installed μ-Pnet monitoring system in a distribution network.

4 Case Study: Event Detection in Power Grid using Advanced Sensing Technology

This section is devoted to the verification of the proposed non-parametric model and the RBCD algorithms. A cross-validation based procedures for the choice of model hyper-parameters is also included. Overall, we will demonstrate, through a real-world application to power system data, that the proposed multiple time series analysis tools enables the discovery of network level outliers that may otherwise be ignored by traditional single time series analysis methods. More case setup details and comparison to other multiple time series methods can be found in supplementary material. The RBCD is very easy to implement, and we provide a Python version one the authors' web-page.

4.1 Data Collection from a Power Grid

The data-set used in this section was collected from a power distribution system equipped in Alameda, CA, with advanced smart meters called phasor measurement units (PMUs) (Von Meier et al. 2014) (Figure 1). Each channel of a particular PMU generates a time series by measuring one type of system state at a certain node. In this experiment, five PMUs are installed at different locations in a distribution subsystem, providing measurements of voltage/current magnitude and phase angle at a high sampling rate. Since all PMUs are connected with one another through the underlying power distribution network, the measurements also demonstrate non-negligible inter-series correlations, in particular for times series generated from the same branch of the network.

All measurements from the PMU netowrk are GPS time stamped to provide time-synchronized observability. The smart meters used in this project provide three-phase voltage and current magnitude and phase angle with 20 seconds time resolution. Measurement data is collected during the period June 02 to July 11, 2015. Each sample is a 60 dimensional vector containing 12 channels per μPMU measuring three phase voltage/current magnitude/angle. Thus for the mutiple time series model, the observed measurement X is $60 \times T$, and the empirical correlation matrix C has dimension 60×60.

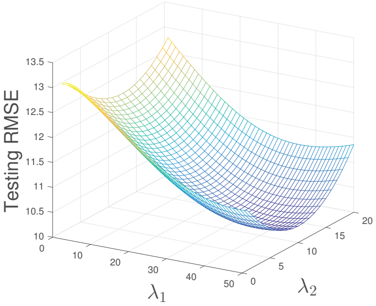

Figure 2: Testing RMSE vs. hyperparameters

4.2 Choice of Hyper-Parameters and the Computational Cost

The proposed non-parametric method has two hyperparameters λ_1 and λ_2, which are weights for temporal and inter-series smoothness, respectively. These hyperparameters determines the complexity of the learned model, and are critical for the performance of the two method. In the sequel we discuss the choice of hyperparameters within a cross validation (CV) framework. First of all, a clean chunk of the multiple time series data[2] is randomly divided into training and testing sets. Let $B \in \mathbb{R}^{M \times T}$ be the indicator matrix having the same dimension as the data matrix X, i.e., $B_{ij} = 1$ if X_{ij} belongs to the training set, and $B_{ij} = 0$ if X_{ij} is assigned to the testing set. Each entry of B follows a Bernoulli distribution Ber(0.7), i.e., we use approximately 70% of the data for training and leave 30% for testing.

Fortunately, the proposed methods are readily amendable to handle missing values (the data points held out for testing): One can simply ignore the loss terms of the testing data points in the first part of (5), or more compactly, use $X \circ B$ to replace the data matrix. To evaluate the CV performance, we use the root mean square error (RMSE) on the testing data set. Figure 2 shows the impact of the two hyperparameters, λ_1 and λ_2, on the testing RMSE of the nonparameteric method. The 2D surface reaches a minimum when $\lambda_1 = 39$ and $\lambda_2 = 10$, demonstrating a trade-off between training fitness and smoothness (complexity). Based on that, we set the two weights accordingly for the nonparametric method.

We also compare the computational cost of RBCD for the non-parametric method and our major competitor, Facets (Cai et al. 2015), which uses EM algorithm for updating contextual HMMs (hence is denoted by EM-CHMM). In addition, the classical mARIMA and signal-series non-parametric model are included into the comparison. To further justify the benefit of RBCD, we include the popular Alternating Direction Method of Multipliers (ADMM) algorithm (Boyd et al. 2011), for the alternative optimization of (5). All numerical experiments are performed on a workstation having dual Xeon5687 CPUs and 72GB memory. The results shown in the sequel are average values of 20 repetitions.

Figure 3 illustrates the required computational time as a function of increasing size of training sequences. Among all

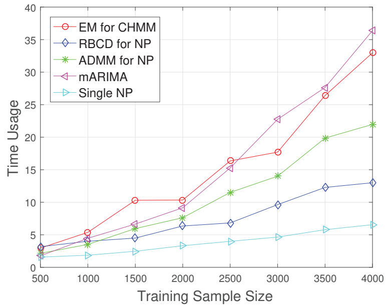

Figure 3: Comparison of Time Usage.

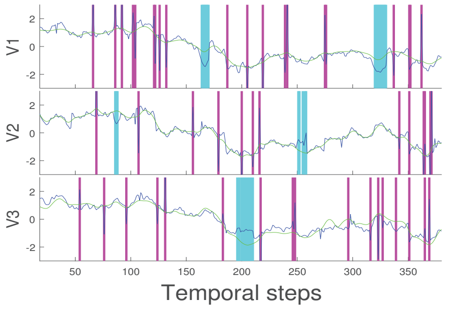

Figure 4: Outlier Detection: the proposed method

methods that incorporate inter-series relatedness, RBCD-NP is the most efficient: For large training size it significantly reduces the running time by at least 42.1% compared to the runner-up ADMM-NP. Although single-NP takes the least time usage, its detection performance is poor and it misses all network level outliers, as will be seen later. It appears that the computational costs of EM-CHMM and mARIMA scale slightly super-linearly and are both much more expensive than that of the non-parametric method.

4.3 Outlier Detection Results

Next we test the proposed methods as a tool for outlier detection. A 120 minute measurement sequence is taken out, which exhibits abnormalities due to sensor or communication failure, and novel events like voltage disturbance due to load changes. For comparison purposes, this data set is also inspected by a power system expert to manually mark outliers and events. We compute an index of novelty by comparing the inferred values $\widehat{X}_{it}$ with the observed values of X_{it} with the absolute distance.

Figure 4 shows the detection results of the proposed nonparametric method. Note that although data from all 60 series are used, only three correlated voltage streams are shown here for clearer presentation. The blue curve in each subplot is the raw data with outliers, and the green curve is the estimated values with the non-parametric method. It is seen that the estimated values are smoothed version of the original data and the measurement noise has been canceled out. For each time series, outliers/novelties are marked with vertical lines when the absolute difference between raw value and estimated

[2] The data used for CV contains very few outliers and is different from the chunk of data used in the next section for validation.

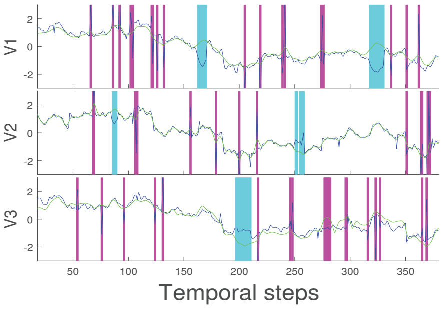

Figure 5: Outlier Detection: CHMM

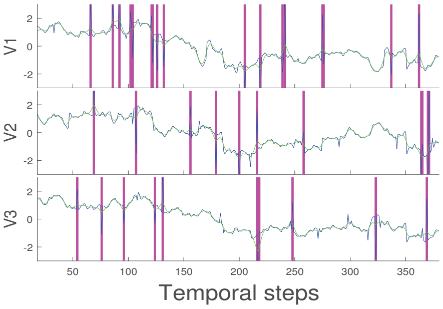

Figure 6: Outlier Detection: spline method

value is larger than 0.73, which is 2σ calculated from all estimation biases.

It appears that our method successfully captured almost all outliers caused by sensor/communication problems or load changes. Those outliers are marked in magenta in each of the panel. More interestingly, due to the incorporation of inter-series dependence, the estimated values for each time series do not always follow its own trend, but are also influenced by other correlated time series. This feature enables the detection of "network level" outliers and novelties, i.e., those data points that significantly violate the correlation structure of the system under measurement. This type of outliers are marked in cyan in each panel of Figure 4. Intuitively, they correspond to power grid events such as three phase imbalance, real and reactive power switching, etc.

To further justify the proposed method and the benefits of incorporating inter-series dependence, we compare them with three alternatives: the Facets with contextual HMMs (CHMM) in Figure 5, the single stream smoothing spline method (Gu 2013) in Figure 6, and the multivariate auto-regressive integrated moving average (mARIMA) model (Box et al. 2015) in Figure 7. Since the detection problem is inherently imbalanced, we report the precision and recall of each method in Table 1, by comparing their detection results with expert labels. More specifically, precision is the fraction of detected events or outliers that are consistent with expert labeling, while recall is fraction of the events or outliers labeled by expert that are successfully retrieved.

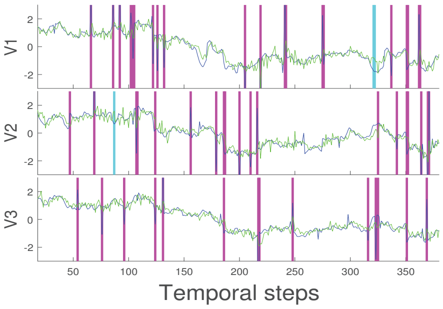

Figure 7: Outlier Detection: m-ARIMA

Table 1: Detection Comparison: Precision and Recall

Method	Simple outliers		network-level events	
	Precison	Recall	Precison	Recall
Propsoed	96.7	91.7	94.8	91.1
CHMM	93.3	90.9	92.5	88.7
s-Spline	84.4	92.8	31.8	19.2
mArima	81.0	85.6	62.4	57.7

At a first glance, the estimated values (green curves) are quite similar to those based on the non-parametric method. In general, CHMM also successfully detects both single stream and network level outliers. However compared to the non-parametric method, CHMM seems to emphasize inter-series relatedness more, while the temporal trend of each series is weighted less. Moreover, the computational cost of our method is much less and scalable than CHMM. Apparently, the single task spline method fails to detect outliers that violate the correlation structure, although in general it provide well-fitted trend for each sequence. The detection results of mARIMA are interesting: due to the non-stationary nature (even after taking difference) of the measurement data, ARIMA model does not provide a good estimation in general. It is observed that some of the single stream outliers were missed, although the method is able to detect several network level outliers.

5 Conclusion and Discussion

To incorporate both temporal dependence and inter-series relatedness for outlier detection, we propose a non-parametric learning method for multiple series, which can be viewed as a multitask version (Pan and Yang 2010) of the classical non-parametric regression method. It is shown that the learning formulation can be extended to handle data with exponential family distribution, and an efficient RBCD algorithm can be use to solve the convex optimization problem. The model and the algorithm is tested with a real-world application, involving outlier detection and event analysis in power distribution networks with high resolution multi-stream measurements. It is shown that the incorporation of inter-series relatedness enables the detection of system level events which would otherwise be unobservable with traditional methods.

The proposed RBCD algorithm bears some interesting features. It resembles SGD in that the expectation of each update is equal to the gradient, while unlike SGD the RBCD ensures a decrease of the objective function in each iteration.

Arguably, recent machine learning literature focuses more on gradient descent based method but block coordinate descent seems to be ignored. This work advocates the use of BCD for a broader class of ML optimization problems.

6 Acknowledgments

This research is funded by the Republic of Singapore National Research Foundation (NRF) through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the UC Berkeley as a center for intellectual excellence in research and education in Singapore.

References

Aggarwal, C. C., and Yu, P. S. 2001. Outlier detection for high dimensional data. In *ACM Sigmod Record*, volume 30, 37–46. ACM.

Aggarwal, C. C., and Yu, P. S. 2008. Outlier detection with uncertain data. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, 483–493. SIAM.

Aggarwal, C. C.; Zhao, Y.; and Philip, S. Y. 2011. Outlier detection in graph streams. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, 399–409. IEEE.

Aggarwal, C. C. 2015. Outlier analysis. In *Data mining*, 237–263. Springer.

Bertsekas, D. P.; Nedi, A.; Ozdaglar, A. E.; et al. 2003. Convex analysis and optimization.

Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3(1):1–122.

Cai, Y.; Tong, H.; Fan, W.; Ji, P.; and He, Q. 2015. Facets: Fast comprehensive mining of coevolving high-order time series. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 79–88. ACM.

Cavraro, G.; Arghandeh, R.; Barchi, G.; and von Meier, A. 2015a. Distribution network topology detection with time-series measurements. In *Innovative Smart Grid Technologies Conference (ISGT), 2015 IEEE Power & Energy Society*, 1–5. IEEE.

Cavraro, G.; Arghandeh, R.; von Meier, A.; and Poolla, K. 2015b. Data-driven approach for distribution network topology detection. *arXiv preprint arXiv:1504.00724*.

Chatterjee, S., and Hadi, A. S. 2015. *Regression analysis by example*. John Wiley & Sons.

Donoho, D. L., and Johnstone, I. M. 1995. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association* 90(432):1200–1224.

Enders, W. 2004. Applied econometric time series, by walter. *Technometrics* 46(2):264.

Fan, J., and Gijbels, I. 1996. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, volume 66. CRC Press.

Gonçalves, A. R.; Von Zuben, F. J.; and Banerjee, A. 2016. Multi-task sparse structure learning with gaussian copula models. *Journal of Machine Learning Research* 17(33):1–30.

Gu, C. 2013. *Smoothing spline ANOVA models*, volume 297. Springer Science & Business Media.

Gupta, M.; Gao, J.; Sun, Y.; and Han, J. 2012. Integrating community matching and outlier detection for mining evolutionary community outliers. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 859–867. ACM.

Gupta, M.; Gao, J.; Aggarwal, C. C.; and Han, J. 2014. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering* 26(9):2250–2267.

Harchaoui, Z., and Lévy-Leduc, C. 2010. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association* 105(492):1480–1493.

Hodrick, R. J., and Prescott, E. C. 1997. Postwar us business cycles: an empirical investigation. *Journal of Money, credit, and Banking* 1–16.

Isermann, R. 2005. Model-based fault-detection and diagnosis–status and applications. *Annual Reviews in control* 29(1):71–85.

Mallat, S. 2008. *A wavelet tour of signal processing: the sparse way*. Academic press.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10):1345–1359.

Von Meier, A.; Culler, D.; McEachern, A.; and Arghandeh, R. 2014. Micro-synchrophasors for distribution systems. In *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*, 1–5.

Zhou, Y., and Spanos, C. J. 2016. Causal meets submodular: Subset selection with directed information. In *Advances in Neural Information Processing Systems*, 2649–2657.

Zhou, Y.; Hu, N.; Spanos, C. J.; et al. 2016. Veto-consensus multiple kernel learning. In *AAAI*, 2407–2414.

Zhou, Y.; Kang, Z.; and Spanos, C. J. 2017. Parametric dual maximization for non-convex learning problems. In *AAAI*.