ONLINE CHURN DETECTION ON HIGH DIMENSIONAL CELLULAR DATA USING ADAPTIVE HIERARCHICAL TREES

Farhan Khan[1], *Ibrahim Delibalta*[2], *Suleyman S. Kozat*[1]

[1]Bilkent University, Ankara, Turkey
[2]AveaLabs, AVEA Iletisim Hizmetleri A.S. Istanbul, Turkey

ABSTRACT

We study online sequential logistic regression for churn detection in cellular networks when the feature vectors lie in a high dimensional space on a time varying manifold. We escape the curse of dimensionality by tracking the subspace of the underlying manifold using a hierarchical tree structure. We use the projections of the original high dimensional feature space onto the underlying manifold as the modified feature vectors. By using the proposed algorithm, we provide significant classification performance with significantly reduced computational complexity as well as memory requirement. We reduce the computational complexity to the order of the depth of the tree and the memory requirement to only linear in the intrinsic dimension of the manifold. We provide several results with real life cellular network data for churn detection.

Index Terms— Churn, big data, online learning, classification on high dimensional manifolds, tree based method.

1. INTRODUCTION

Online data classification is widely investigated in data mining [2], machine learning [4] and churn detection in cellular networks [1]. For applications involving big data [2], when the input vectors are high dimensional, the classification models are computationally complex and usually result in overfitting.

In this paper, we study non-linear logistic regression using high dimensional data assuming the data lies on a time varying manifold. We partition the feature space into several regions to construct a piecewise linear model as an approximation to the non-linearity between the observed data and the desired data. However, instead of fixing the boundaries of the regions, we use the notion of context trees [5, 6] to represent a broad class of all possible partitions for the piecewise linear models. We specifically introduce an algorithm that incorporates context trees [5, 6] for online learning of the high dimensional manifolds and perform logistic regression on big data.

This work has been supported in part by AveaLabs, TUBITAK TEYDEB 1501 no. 3130095 project.

In most modern applications, where high dimensional data is involved, learning and regression on the manifolds are widely investigated. For instance, in cellular networks [1], large amounts of high dimensional, time varying data of various users are used for churn detection. The problem of manifold learning and regression is rather easy when all the data is available in advance (batch), and lies around the same single static submanifold [8]. However, in online manifold learning, it is difficult to track the variation in data because of the high dimensionality and time varying statistical distributions [8]. Hence, we introduce a comprehensive solution that includes online logistic regression on a high dimensional time series.

Note that various approaches are studied for the dimensionality reduction as a preprocessing step for the analysis of the high dimensional data [8,9]. In our approach, however, we use context trees to perform logistic regression, which adapts automatically to the intrinsic low dimensionality of the data by maintaining the "geodesic distance" [7] while operating on the original regressor space. In the domain of online nonlinear regression, context trees have been used to partition the regressor space hierarchically, and to construct a competitive algorithm among a broader class of algorithms [6]. However, we use hierarchical tree structure to track and learn the manifold in a high dimensional setting. In addition to solving the problem of high dimensionality by incorporating manifold learning, our algorithm also performs online logistic regression.

To this end, we introduce an algorithm that uses a tree structure to hierarchically partition the high dimensional feature space. We extend the algorithm by incorporating approximate Mahalonobis distance as in [8] to adapt the feature space to its intrinsic lower dimension. Our algorithm also adapts the corresponding regressors in each region to minimize the final regression error. We show that our methods are truly sequential and generic in the sense that they are independent of the statistical distribution or structure of the data or the underlying manifold. In this sense, the proposed algorithms learn $i)$ the structure of the manifolds, $ii)$ the structure of the tree, $iii)$ the low dimensional projections in each region, $iv)$ the logistic regression parameters in each region, and $v)$ the linear combination weights of all possible partitions, to mini-

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

mize the final regression error.

The paper is organized as follows. In Section 2, we formally describe the problem setting in detail. In Section 3, we extend the context tree algorithm [6] to the high dimensional case and describe the tools we use such as approximate Mahalonobis distance, and define our parameters. Then, we formally propose our algorithm. In Section 4, we perform logistic regression on the real life cellular network data for churn detection using the proposed algorithm. We compare the performance of our algorithm with well known algorithms in the literature such as linear discriminant analysis [4], support vector machines (SVM) [4, 10] and classification trees [4], using computational complexity and success rate.

2. PROBLEM DESCRIPTION

All vectors used in this paper are column vectors, denoted by boldface lowercase letters. Matrices are denoted by boldface uppercase letters. For a vector $\mathbf{v}$, $\|\mathbf{v}\|^2 = \mathbf{v}^T\mathbf{v}$ is squared Euclidean norm and $\mathbf{v}^T$ is the ordinary transpose. $\mathbf{I}_k$ represents a $k \times k$ identity matrix.

We investigate online logistic regression using high dimensional data, i.e., when the dimension of data $D \gg 1$. We observe a desired label sequence $\{y[n]\}_{n\geq 1}$, $y[n] \in \{-1, 1\}$, and regression vectors $\{\mathbf{x}[n]\}_{n\geq 1}$, $\mathbf{x}[n] \in \mathbb{R}^D$, where D denotes the *ambient dimension*. The data $\mathbf{x}[n]$ are measurements of points lying on a submanifold $S_{m[n]}$, where the subscript $m[n]$ denotes the time varying manifold, i.e. $\mathbf{x}[n] \in S_{m[n]}$. The *intrinsic dimension* of the submanifolds $S_{m[n]}$ are d, where $d \ll D$. The submanifolds $S_{m[n]}$ can be time varying. At each time n, a vector $\mathbf{x}[n]$ is observed. Then $z[n]$ is given by:

$$z[n] = f_n(\mathbf{x}[n]), \tag{1}$$

where $f_n(.)$ is a non-linear, time varying function and $z[n] = 0$ is a separating hyperplane between the two classes. The instantaneous regression error is given by: $e[n] = y[n] - z[n]$. The estimate of the desired label is calculated by the following logistic function,

$$\hat{y}[n] = h(z[n]), \tag{2}$$

where $h(.)$ is a signum function, i.e.,

$$h(z) = \begin{cases} 1, & \text{if } z \geq 0, \\ -1, & \text{if } z < 0, \end{cases} \tag{3}$$

We approximate the nonlinear function $f_n(.)$ by piecewise linear models such that the $\mathbb{R}^D$ regressor space is divided into various regions. We assume that there is a linear relationship between $x[n]$ and $z[n]$ in each region. We use a hierarchical tree structure that partitions the regressor space into various regions. We define a "partition" of the D-dimensional regressor space as a specific partitioning $\mathcal{P}_i = \{R_{i,1}, ..., R_{i,J_i}\}$, where $\bigcup_{j=1}^{J_i} R_{i,j} = \mathcal{R}$, $R_{i,j}$ is a region in the D-dimensional regressor space and $\mathcal{R} \in \mathbb{R}^D$ is the complete D-dimensional regressor space. In general, for a tree of depth K, there are as many as 1.5^{2^K} possible partitions as shown in Fig. 1. Each of these doubly exponential number of partitions can be used to construct a piecewise linear model. For instance, for a j^{th} region $\mathcal{R}_j$ of partition $\mathcal{P}_i$, the estimate of $y[n]$ is given by,

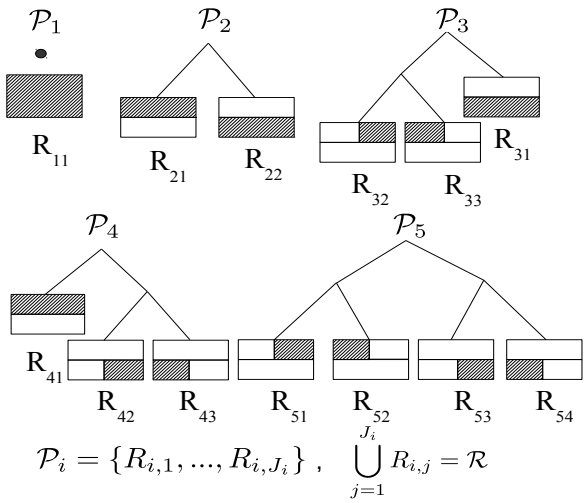

Fig. 1: A full tree of depth 2 that represents all possible partitions of the two dimensional space, $\mathcal{P} = \{\mathcal{P}_1, ..., \mathcal{P}_{N_K}\}$ and $N_K \approx (1.5)^{2^K}$, where K is the depth of the tree. Here $N_K = 5$.

$$\hat{y}_j[n] = h(\mathbf{w}_j[t]^T\mathbf{x}[n] + b_j[t]]), \tag{4}$$

where $\mathbf{w}_j[t]$ is the weight vector in $\mathbb{R}^D$, $j \in \{1, ..., J_i\}$, J_i is the number of regions the feature space is divided into by the partition $\mathcal{P}_i$, $b_j[t] \in \mathbb{R}$ denotes an offset, $t = \{n; \mathbf{x}[n] \in \mathcal{R}_j\}$. The overall estimate of $y[n]$ by the partition $\mathcal{P}_i$ is then given by:

$$\hat{y}_{\mathcal{P}_i}[n] = h(\mathbf{w}_j\mathbf{x}[n] + b_j), \tag{5}$$

where $j \in \{1, ..., J_i\}$, $\mathbf{x}[n] \in \mathcal{R}_j$.

However, instead of the doubly exponential number of possible partitions, we use the context tree algorithm [6] that competes with the class of all possible partitions with a computational complexity only linear in the depth of the tree.

In Fig. 2, a context tree of depth $K = 2$, that partitions the $\mathbb{R}^2$ space into at most four possible regions, is shown. Here, we specifically minimize the following regret over any n [6],

$$\sum_{t=1}^{n}(y[t] - \hat{y}_q[t])^2 - \inf_{\mathcal{P}_i}\sum_{t=1}^{n}(y[t] - \hat{y}_{\mathcal{P}_i}[t])^2, \tag{6}$$

where $\hat{y}_{\mathcal{P}_i}[t]$ is the estimate of $y[t]$ by the partition $\mathcal{P}_i, i \in \{1, ..., 1.5^{2^K}\}$, and $\hat{y}_q[t]$ is the estimate of $y[t]$ by the context tree algorithm [6].

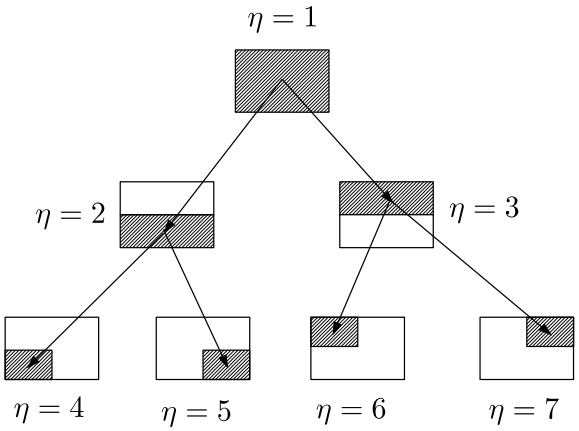

Fig. 2: A two dimensional context tree of depth 2

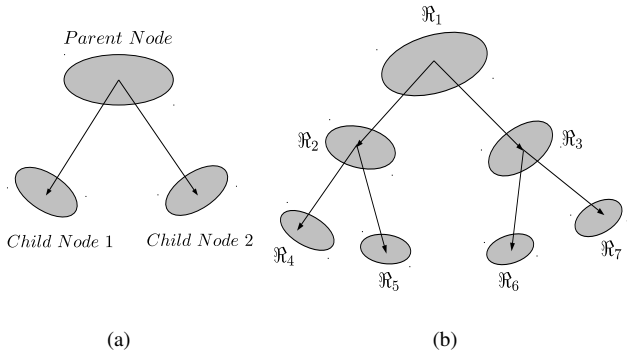

Fig. 3: (a) A parent node and its two children in an adaptive hierarchical tree. Each node (subset) is a two dimensional ellipsoid defined by its parameters $\{\mathbf{Q}_\eta[n], \mathbf{\Lambda}_\eta[n], \mathbf{c}_\eta[n]\}$. (b) A dynamic hierarchical tree of depth K where each η represents a subset defined by (7) and $\eta \in \{1, 2, ..., 2^{K+1} - 1\}$

In our current setting, where the feature vectors lie on a high dimensional manifold and only a subset of features are observed, we use the projections of the input vectors on the intrinsic dimension of the manifolds instead of the original feature vectors. We propose an algorithm that uses a hierarchical tree structure to track the underlying manifold and perform logistic regression on the projected $\mathbb{R}^d$ space, where d is the intrinsic dimension of the manifold and $d \ll D$.

3. LOGISTIC REGRESSION ON MANIFOLDS

To escape the curse of dimensionality, we perform regression on high dimensional data by mapping the regressor vectors to low dimensional projections. We assume that the observed data $\mathbf{x}[n] \in \mathbb{R}^D$ lies on time varying submanifolds $S_{m[n]}$. We can solve the problem of non-linear regression by using piecewise linear modeling as explained in Section 2, where the regressor space, i.e., $\mathbb{R}^D$ can be partitioned into several regions. However, in the new setting, since the data lies on submanifolds with lower intrinsic dimension, we use the lower dimensional projections instead of the original $\mathbb{R}^D$ regressor space. We define the piecewise regions in $\mathbb{R}^d$ for each node that correspond to the low dimensional submanifolds. However, since the submanifolds are time varying, the regions are not fixed. We define these regions by the subsets [8, 9]:

$$\begin{aligned}\Re_j[n] = \{\mathbf{x}[n] \in \mathbb{R}^{\mathrm{D}} : \mathbf{x}[n] = \mathbf{Q}_j[n]\boldsymbol{\beta}_j[n] + \mathbf{c}_j[n], \\ \boldsymbol{\beta}_j^T[n]\mathbf{\Lambda}_j^{-1}[n]\boldsymbol{\beta}_j[n] \leq 1, \boldsymbol{\beta}_j[n] \in \mathbb{R}^d\},\end{aligned} \tag{7}$$

where each subset $\Re_j[n]$ is a $d-$dimensional ellipsoid assigned to each node of the tree. The matrix $\mathbf{Q}_j[n] \in \mathbb{R}^{\mathrm{D}\times\mathrm{d}}$ is the subspace basis in $d-$dimensional hyperplane and the vector $\mathbf{c}_j[n]$ is the offset of the ellipsoid from the origin. The matrix $\mathbf{\Lambda}_\eta[n] \triangleq \mathrm{diag}\{\lambda_\eta^{(1)}[n], ..., \lambda_\eta^{(d)}[n]\}$ with $\lambda_\eta^{(1)}[n] \geq ... \geq \lambda_\eta^{(d)}[n] \geq 0$, contains the eigen-values of the covariance matrix of the data $\mathbf{x}[n]$ projected onto each hyperplane. The subspace basis $\mathbf{Q}_\eta[n]$ specifies the orientation or direction of the hyperplane and the eigen-values specify the spread of the data within each hyperplane [8, 9]. The projections of $\mathbf{x}[n]$ on the basis $\mathbf{Q}_\eta[n]$ are used as new regression vectors, $\boldsymbol{\beta}_\eta[n] = \mathbf{Q}_\eta^T[n](\mathbf{x}[n] - \mathbf{c}_\eta[n])$. We represent these regions by a tree structure given in Fig. 3(b).

The piecewise regions are now $d-$dimensional ellipsoids instead of the original $\mathbb{R}^D$ space, therefore, we use the approximate Mahalonobis distance D_M [8] to measure the closeness of the leaf nodes with the input vector $\mathbf{x}[n]$, i.e.,

$$\eta^* = \arg\min_\eta D_M(\mathbf{x}[n], \Re_\eta[n]). \tag{8}$$

Once the minimum distance node, η^* is known, we then use the projection of $\mathbf{x}[n]$ on its subspace to calculate the new regressor vector $\boldsymbol{\beta}_{\eta^*}[n]$. We use this η^* as the $\{K+1\}^{th}$ dark node in the context tree algorithm [6] and the rest of K dark nodes are the ancestor nodes of η^* till the root node $\eta = 1$. We use each of these dark nodes to estimate $z[n]$ as $\tilde{z}_k[n] = \mathbf{w}_k[n-1]^T\boldsymbol{\beta}_k[n] + b_k[n-1]$, and linearly combine these estimates as follows,

$$\hat{z}[n] = \sum_{k=1}^{K+1} \mu_k[n-1]\tilde{z}_k[n], \tag{9}$$

where the combination weights are determined by the performance of the k^{th} dark node in the past till $n-1$ [6]. We assume linear discriminants in each region and learn the weight vectors $\mathbf{w}_k[n-1] \in \mathbb{R}^d$ and the offsets $b_k[n-1] \in \mathbb{R}$ using the Recursive Least Square (RLS) algorithm [3].

When a new data sample $\mathbf{x}[n]$ arrives, we first calculate the minimum distance node among the leaf nodes of the tree according to (8). We mark this node as the $\{K+1\}^{th}$ dark node and determine the rest of the dark nodes by climbing

Algorithm 1: Logistic Regression using AHT

for $\eta = 1$ to $2^{K+1} - 1$; **do**
  Initialize parameters $\{\mathbf{Q}_\eta, \mathbf{\Lambda}_\eta, \mathbf{c}_\eta, \delta_\eta\}$,
  Initialize the logistic regression parameters $\{\boldsymbol{\omega}_\eta, \theta_\eta\}$,
  Initialize the performance measure C_η,
end for
for $k = 1, ..., K+1$; **do**
  $\mu_k[0] = 0, \sigma_k[0] = 0$
end for

Algorithm:

for $n = 1, ...$ **do**
  for $\mathbf{x}[n]$, find the minimum distance node among the leaf nodes, i.e., $\{K+1\}^t h$ dark node
  Find the ancestor nodes of the $\{K+1\}^{th}$ dark node
  Project $\mathbf{x}[n]$ on each dark node and determine $\boldsymbol{\beta}_k[n]$ for $k \in \{1, ..., K+1\}$
  Estimate the desired label using (10)
  Update the subset, combination weights and regressor parameters for each dark node
end for

up the tree till the root node [**?**, **?**]. We then project the observed data sample on each of these dark nodes, train a linear discriminant by using the projection as the new regressor vector, and estimate the desired data label $y[n]$ as $\tilde{z}_k[n]$, for $k \in \{1, ..., K+1\}$. We use (9) and the logistic function in (3) to estimate the desired label, i.e.,

$$\hat{y}[n] = h\left(\sum_{k=1}^{K+1} \mu_k[n-1]\tilde{z}_k[n]\right) \tag{10}$$

We next update the subsets parameters $\{\mathbf{Q}_k[n], \mathbf{\Lambda}_k[n], \mathbf{c}_k[n], \delta_k[n]\}$ belonging to each dark node [8, 9] using the update step in the Adaptive Hierarchical Tree algorithm in [**?**, **?**]. The subset basis $\mathbf{Q}_k[n]$ is updated using PeTReLS-FO algorithm [8, 11]. We also update the combination weights $\mu_k[n]$ according to the performance of the node k in estimating the desired data. Moreover, we update the logistic regressor parameters by using the RLS algorithm.

4. EXPERIMENTS

In this section, we use the proposed adaptive hierarchical trees algorithm described in Section 3 for the online churn detection using high dimensional cellular network data of $N = 10,000$ users. The original dimension of the regressor vectors $\mathbf{x}[n]$ is $D = 164$ and the desired labels $y[n] \in \{-1, 1\}$ where $y[n] = -1$ for "no churn" and $y[n] = 1$ for "churn". The dataset contains missing values and we observe a subset of the D features for each user. The dataset is also imbalanced as the "no churn" class contains approximately 77% of samples. Therefore, we perform oversampling on the minority class to balance the dataset [4]. We perform online logistic regression choosing $d \in \{10, 20, 30, 40, 50, 60\}$. We plot the success rate versus d for $K \in \{1, 2, 3\}$ in Fig. 4.

Table 1: Success rate, false alarm rate and detection rate for AHT ($d = 40$, $K = 3$), SVM, Classification Trees and AHT-SVM

Algorithm	Sucess Rate	False Alarm Rate	Detection Rate
AHT	0.91	0.04	0.93
SVM	0.985	0.03	0.97
CTrees	0.98	0.04	0.95
AHT-SVM	0.99	0.01	0.99

The logistic regression in $\mathbb{R}^d$ shows great improvement in performance for churn detection in high dimensional cellular data for $d \ll D$. In Fig. 4, we show that choosing a larger d improves the performance of the algorithm, however, it reaches saturation and further increasing d results in overfitting and increased complexity. Hence, using only $d = 40$ features instead of $D = 164$, we achieve a success rate of over 90% with simple linear discriminant functions within each piecewise region.

It is interesting to note that for smaller K, the algorithm performs worse for $d < 40$, however, it reaches the same success rate as $K > 2$ as we choose $d > 40$. We also perform logistic regression on the original $\mathbb{R}^D$ feature vectors using online linear discriminants and the success rate is as low as 32%. Therefore, the proposed algorithm not only reduces the computational complexity but also produces outstanding success rate.

We next compare the performance of our algorithm with well known classification algorithms, i.e., SVM and classification trees while using these algorithms offline. We show that our algorithm produces comparable results to batch SVM and classification trees with much less computational and time complexity as our algorithm uses linear discriminants in the online setting.

The adaptive hierarchical trees (AHT) algorithm for logistic regression uses ensemble learning [4] as it linearly combines several linear discriminants and produces outstanding results with much less complexity than SVM and Classification trees (CTrees). We also use a combination of our algorithm and SVM by using SVM within each piecewise region instead of linear discriminants (AHT-SVM) and achieved 99% success rate with a false alarm rate $= 0.01$. However, this approach is much complex than the original AHT and is not suitable for online learning. The results are shown in Table 1.

5. CONCLUSION

We consider the problem of logistic regression on high dimensional data using piecewise linear discriminants. We assume that the feature vectors lie on a high dimensional and time varying submanifolds. We propose an algorithm that effectively learns the underlying structure of the manifold and performs piecewise logistic regression on the low dimensional projections. We use hierarchical tree structure for the piece-

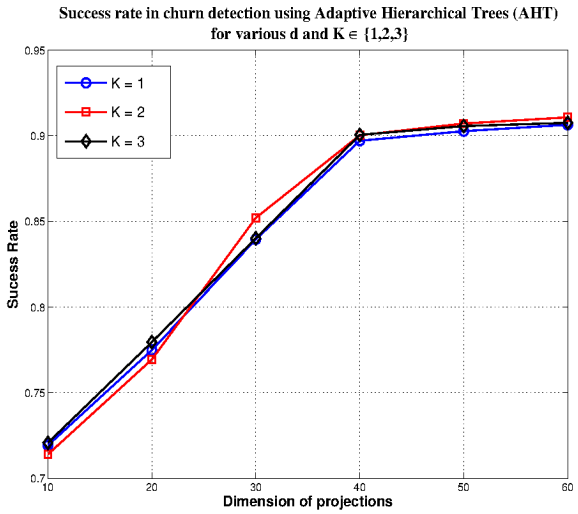

Fig. 4: Success rate in Churn detection using Adaptive hierarchical Trees algorithm for logistic regression with various choices of d and K

wise modeling and linear discriminants within each region. We achieve comparable results to other classification algorithms such as SVM and classification trees with much less complexity and memory requirement while working in an online setting. We use the proposed algorithm on high dimensional cellular data for churn detection and achieve a detection rate of 0.93 with a false alarm rate of only 0.04.

REFERENCES

[1] J. A. Deri, J. M. F. Moura, "Churn detection in large user networks ," *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* 2014.

[2] X. Wu, X. Zhu, G.-Q. Wu, W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[3] A. H. Sayed, *Fundamentals of Adaptive Filtering*. NJ: John Wiley & Sons, 2003.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] F. M. J. Willems,"The context-tree weighting method: extensions," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 792–798, 1998.

[6] S. S. Kozat, A. C. Singer, and G. C. Zeitler, "Universal piecewise linear prediction via context trees," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3730–3745, 2007.

[7] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*. Springer, 1997.

[8] Y. Xie, J. Huang and R. Willett, "Change-point detection for high-dimensional time series with missing data," *IEEE Journal of Selected Topics in Signal Processing*,vol. 7, no. 1, pp. 12–27, 2013.

[9] Y. Xie and R. Willett, "Online logistic regression on manifolds," *IEEE Internation Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[10] F. Khan, D. Kari, A. Karatepe and S. .S. Kozat, "Universal Nonlinear Regression on High Dimensional Data Using Adaptive Hierarchical Trees," *IEEE Transactions on Big Data*, vol. PP, no. 99, pp. 1–1, 2016.

[11] F. Khan, I. Delibalta and S. .S. Kozat, "High dimensional sequential regression on manifolds using adaptive hierarchical trees," *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2015.

[12] N. Cristianini and J. Shawe-Taylor, *Support Vector Machines and Other Kernal-Based Learning Methods*. Cambridge University Press, 2000.

[13] Y. Chi, Y. C. Eldar, and R. Calderbank, "PETReLS: Subspace estimation and tracking from partial observations," *Int. Conf. on Acoustic, Speech, and Sig. Processing*, March 2012.