AN EFFICIENT MIXTURE OF EXPERTS METHOD FOR BIG DATA APPLICATIONS

O. Fatih Kilic[⋆] *M. Omer Sayin*[†] *Suleyman S. Kozat*[⋆]

[⋆] Department of Electrical and Electronics Engineering
Bilkent University, Ankara 06800, Turkey
[†]School of Electrical and Computer Engineering
University of Illinois at Urbana Champaign, Urbana, Illinois

ABSTRACT

We introduce a new combination approach for the mixture of adaptive filters based on the set-membership filtering (SMF) framework. We perform SMF to combine the outputs of several parallel running adaptive algorithms and propose unconstrained, affinely constrained and convexly constrained combination weight configurations. Here, we achieve better trade-off in terms of the transient and steady-state convergence performance while providing significant computational reduction. Hence, through the introduced approaches, we can greatly enhance the convergence performance of the constituent filters with a slight increase in the computational load. In this sense, our approaches are suitable for big data applications where the data should be processed in streams with highly efficient algorithms. In the numerical examples, we demonstrate the superior performance of the proposed approaches over the state of the art using the well known datasets in the machine learning literature.

***Index Terms*—** Big Data, mixture approach, set-membership filtering, computational reduction

1. INTRODUCTION

Recently, the mixture approaches have been proposed to combine various adaptive filters with different configurations to achieve better performance than any of the individual algorithm in the mixture [1–6]. Particularly, through the mixture approach we can achieve enhanced performance in a wide range of adaptive filtering applications.

However, we emphasize that the mixture approaches multiplicatively increase the combination load, since they need to run several adaptive algorithms in parallel. Hence, these approaches may not be suitable for applications involving big data due to their impractical computational need.

To this end, in this paper, we introduce a new mixture approach based on set-membership filtering (SMF) framework introduced by Gollamudi et. al. [7] We developed unconstrained, affine and convex combination methods using SMF in order to reduce computational load and achieve improved performance in mixture approaches [8–10]. In the conventional least squares algorithms, e.g., the LMS algorithm (or the stochastic gradient descent algorithm), we minimize a cost function of the error term defined as the difference between the desired and the estimated signals. On contrary, the set membership filtering approach seeks to find any parameter yielding smaller error terms than a predefined bound. SMF approach achieves relatively fast convergence performance in addition to the reduced computational load since we do not update the parameter unless we obtain larger error than the bound [11, 12].

This work is in part supported by Turkish Academy of Science Outstanding Researcher Programme and Tubitak Contract No: 113E517.

2. PROBLEM DESCRIPTION

Considering an on-line setting where only the current feature vector [1] $\mathbf{x}(t)$ at time $t \geq 1$ is available for modeling the corresponding data $d(t)$, our aim is to sequentially estimate $d(t)$ such that $\hat{d}(t) = f(\mathbf{x}(t))$ and for the estimation, we use linear mixture of parallel adaptive filters.

In this structure, our system consists of two parts. In the first part we have m adaptive filters running in parallel to estimate the desired signal $d(t)$ as in Fig.1. Each filter with their parameter vector $\mathbf{w}_i(t)$, $i = 1, \cdots, m$ and the input vector $\mathbf{x}(t)$ produces an estimate $\hat{d}_i(t) = \mathbf{x}^T(t)\mathbf{w}_i(t)$ and in next step we update their parameter vector according to their estimation error $e_i(t) \triangleq d(t) - \hat{d}_i(t)$.

In the second part of the system, we have the mixture stage. At this point, we obtain the final estimate of the system by linearly combining the estimates of the parallel adaptive filters as $\hat{d}(t) = \mathbf{w}^T(t)\mathbf{y}(t)$ where $\mathbf{y}(t) = col\{\hat{d}_1(t), \cdots, \hat{d}_m(t)\}$ is constituent estimates vector and $\mathbf{w}(t) = col\{w^{(1)}(t), \cdots, w^{(m)}(t)\}$ is mixture weights vector. Linear combination parameters of this stage are updated adaptively according to the final estimation error $e(t) \triangleq d(t) - \hat{d}(t)$.

Use of conventional least squares algorithms such as LMS algorithm in these mixture combination systems results in the update of parameter vectors of constituent filters at each step.

[1]Through this paper, bold lower case letters denote column vectors and bold upper case letter denote matrices. For a vector $\mathbf{a}$ (or matrix $\mathbf{A}$), $\mathbf{a}^T$ (or $\mathbf{A}^T$) is its ordinary transpose. The operator $\mathrm{col}\{\cdot\}$ produces a column vector or a matrix in which the arguments of $\mathrm{col}\{\cdot\}$ are stacked one under the other. For a given vector $\mathbf{w}$, $\mathbf{w}^{(i)}$ denotes the ith individual entry of $\mathbf{w}$. Similarly for a given matrix $\mathbf{G}$, $\mathbf{G}^{(i)}$ is the ith row of $\mathbf{G}$. For a vector argument, $\mathrm{diag}\{\cdot\}$ creates a diagonal matrix whose diagonal entries are elements of the associated vector.

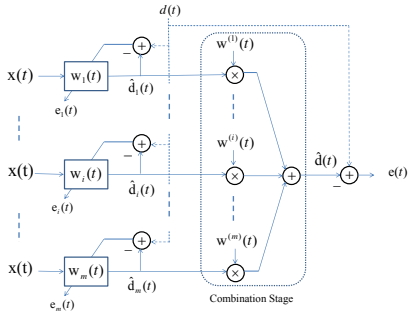

Fig. 1: Mixture combination of parallel filters

This notion may not be desirable for most big data applications due to high computational load that these updates will create. Therefore, as a solution, we employ set membership filters and their mixture combination for this structure.

In subsequent sections, we first introduce the structure of the SMF, then we introduce methods for linear mixture combination of these set membership filters.

3. STRUCTURE OF SET-MEMBERSHIP FILTERS

For the general linear-in-parameter filters whose input is $\mathbf{x} \in \mathbb{R}^n$, the desired output is real scalar d and the output of the filter is $\hat{d} = \mathbf{x}^T\mathbf{w}$ where $\mathbf{w} \in \mathbb{R}^n$ is the parameter vector for the filter. The filter error is defined as $e(\mathbf{w}) = d - \hat{d}$. In a conventional least squares algorithm, filter estimates the parameter vector to minimize the cost which is a function of the filter error [13]. However, in the set membership filtering scheme, we update the parameter vector to satisfy a predefined upper bound γ on the filter error for all data pairs $(d, \mathbf{x})$ in a model space $\mathcal{S}$ such that $|e(\mathbf{w})|^2 \leq \gamma,\ \forall (d, x) \in \mathcal{S}$. Therefore any parameter vector satisfying this condition is an acceptable solution and the set of these solutions forms the feasibility set which is defined as

$$\Gamma \triangleq \bigcap_{(d,\mathbf{x})\in\mathcal{S}} \{\mathbf{w} \in \mathbb{R}^n : |d - \mathbf{x}^T\mathbf{w}|^2 \leq \gamma^2\}. \tag{1}$$

If the model space $\mathcal{S}$ is known priorly, then it is possible to estimate the feasibility set or a parameter vector in it. However, there is no closed form solution for an arbitrary $\mathcal{S}$ and in practice the model space is not known completely or it is time-varying [7]. Therefore we estimate the feasibility set or one of its members using set-membership adaptive recursive techniques (SMART).

Considering a practical case, where only measured data pair $(d_t, \mathbf{x}_t) \in \mathcal{S}$ is available, the constraint set $\mathcal{H}_t$ containing all parameter vectors satisfying error bound condition is defined as $\mathcal{H}(t) \triangleq \{\mathbf{w} \in \mathbb{R}^n : |d(t) - \mathbf{w}^T\mathbf{x}(t)| \leq \gamma\}$ and an estimate for the feasibility set at time t is membership set $\phi_t \triangleq \bigcap_{\tau=1}^{t} \mathcal{H}(\tau)$. We approximate the membership set for tractable and computable results by projecting current parameter vector $\mathbf{w}(t)$ onto constraint set $\mathcal{H}(t+1)$ if it is not contained in it and assure an error upper bound of γ [7].

We express the problem defined above as

$$\mathbf{w}(t+1) = \arg \min_{\mathbf{w}\in\mathcal{H}(t+1)} \|\mathbf{w} - \mathbf{w}(t)\|^2. \tag{2}$$

Solution to problem in (2) is

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu(t)\frac{\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{x}(t)} \tag{3}$$

where

$$\mu(t) = \begin{cases} 1 - \frac{\gamma}{|e(t)|} & \text{if } |e(t)| > \gamma, \\ 0 & \text{otherwise.} \end{cases}$$

The resulting algorithm in (3) is named as set membership normalized least mean square algorithm (SM-NLMS) and achieves better convergence speed and steady-state MSE with reduced computational load than NLMS algorithm [7]. In the next section, we use this SMF structure in constituent and combination filters of mixture combination approach to create computationally efficient and fast converging estimation system.

4. PROPOSED COMBINATION METHODS

We employ SMF scheme for the mixture combination of constituent set-membership filters with different error bounds running in parallel to estimate the desired signal $d(t)$. We use a system where m SMF filter running in parallel as in Fig.1, each one updates their parameter vector $\mathbf{w}_i(t) \in \mathbb{R}^n$ and produces estimate $\hat{d}_i(t) = \mathbf{x}^T(t)\mathbf{w}_i(t)$ with respect to its bound γ_i. In the combination stage of m constituent filters, we combine each filter output linearly through time variant weight vector $w(t)^{(i)} \in \mathbb{R}^m$ which is trained with combining SMF filter with bound $\bar{\gamma}$. We denote input to the combination stage as $\mathbf{y}(t) \triangleq col\{\hat{d}_1(t), ..., \hat{d}_m(t)\}$ and the parameter vector of the combination stage is $\mathbf{w}(t) \triangleq col\{w^{(1)}(t), ..., w^m(t)\}$. The output of the combination stage is $\hat{d}(t) = \mathbf{y}^T(t)\mathbf{w}(t)$ and the final estimation error is $e(t) \triangleq d_t - \hat{d}(t)$.

In the following subsections, we seek and train parameter vectors for the combination stage weights satisfying upper bound $\bar{\gamma}$ within different parameter spaces.

4.1. Unconstrained Linear Mixture Parameters

The first parameter space is for the unconstrained linear mixture weights and defined as $\mathcal{W}_1 \triangleq \{\mathbf{w} \in \mathbb{R}^m\}$ which is the Euclidean space. Therefore, within the SMF scheme, for finding and update of the weights we have

$$\mathbf{w}(t+1) = \arg \min_{\mathbf{w}\in\mathcal{H}_1(t)} ||\mathbf{w} - \mathbf{w}(t)||^2 \tag{4}$$

where $\mathcal{H}_1(t) \triangleq \{\mathbf{w} \in \mathcal{W}_1 : |d(t) - \mathbf{w}^T\mathbf{y}(t)| \leq \bar{\gamma}$ is the constraint set for the update and the solution for the (4) as we did in (2) yields

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu(t)\frac{\mathbf{y}(t)e(t)}{\mathbf{y}^T(t)\mathbf{y}(t)} \tag{5}$$

where

$$\mu(t) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(t)|} & \text{if } |e(t)| > \bar{\gamma}, \\ 0 & \text{otherwise.} \end{cases}$$

4.2. Affine Mixture Parameters

Parameter space for the affine mixture weights is defined as $\mathcal{W}_2 \triangleq \{\mathbf{w} \in \mathbb{R}^m : \mathbf{1}^T\mathbf{w} = 1\}$ where $\mathbf{1} \in \mathbb{R}^m$ denotes

a vector of ones such that sum of weights to be one, i.e. $\sum_{i=1}^{m} w^{(i)} = 1$. Therefore, the constraint set in this case is $\mathcal{H}_2(t) \triangleq \{\mathbf{w} \in \mathcal{W}_2 : |d(t) - \mathbf{w}^T\mathbf{y}(t)| \leq \bar{\gamma}\}$. We remove the affine constraint with the following parametrization. Define parameter vector $\mathbf{z}(t) \in \mathbb{R}^{n-1}$ where

$$\mathbf{z}^{(i)}(t) \triangleq \mathbf{w}^{(i)}(t),\ \forall i \in \{1, 2, ..., m-1\}$$

and

$$\mathbf{w}^{(m)}(t) = 1 - \sum_{i=1}^{m-1} \mathbf{z}^{(i)}(t) \tag{6}$$

Here in (6) we present $z(t)$ as the unconstrained parameter vector and define $a(t)$ as the desired signal and $c(t)$ as the input to the unconstrained optimization problem which is given as

$$\mathbf{z}(t+1) = \arg \min_{\mathbf{z} \in \widetilde{\mathcal{H}}_2(t)} \|\mathbf{z} - \mathbf{z}(t)\|^2, \tag{7}$$

where the constraint set is defined as $\widetilde{\mathcal{H}}_2(t) \triangleq \{\mathbf{z} \in \mathbb{R}^{m-1} : |a(t) - \mathbf{z}^T\mathbf{c}(t)| \leq \gamma\}$ and we can express

$$a(t) = d(t) - \hat{d}_m(t);\ c(t) = \begin{bmatrix} \hat{d}_1(t) - \hat{d}_m(t) \\ \vdots \\ \hat{d}_{m-1}(t) - \hat{d}_m(t) \end{bmatrix}^T.$$

Since now the optimization problem is same as in unconstrained case, as in (4) the solution yields

$$\mathbf{z}(t+1) = \mathbf{z}(t) + \mu(t)\frac{\mathbf{c}(t)e(t)}{\mathbf{c}(t)^T\mathbf{c}(t)} \tag{8}$$

where

$$\mu(t) = \begin{cases} 1 - \frac{\gamma}{|e(t)|} & \text{if } |e(t)| > \gamma, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore with the relation in (6), we can write for $\mathcal{W}_2$ space that

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu(t)\frac{\mathbf{G}\mathbf{y}(t)e(t)}{\mathbf{y}(t)^T\mathbf{G}\mathbf{y}(t)} \tag{9}$$

where

$$\mathbf{G} \triangleq \begin{bmatrix} \mathbf{I}_{m-1} & -\mathbf{1} \\ -\mathbf{1}^T & m-1 \end{bmatrix}$$

and

$$\mu(t) = \begin{cases} 1 - \frac{\gamma}{|e(t)|} & \text{if } |e(t)| > \gamma, \\ 0 & \text{otherwise.} \end{cases}$$

and $-\mathbf{1} \in \mathbb{R}^{m-1}$ is a vector where all its elements is minus one.

4.3. Convex Mixture Parameters

Finally, the parameter space for the convex mixture weights is defined as $\mathcal{W}_3 = \{\mathbf{w} \in \mathbb{R}^m : \mathbf{1}^T\mathbf{w} = 1 \wedge \mathbf{w}^{(i)} \geq 0, \forall i \in \{1, ..., m\}\}$. In order to get unconstrained optimization problem as we did above, we re-parameterize vector $\mathbf{w}(t)$ with the parameter vector $\mathbf{z}(t) \in \mathbb{R}^m$ as in [8]

$$\mathbf{w}^{(i)}(t) = \frac{e^{-\mathbf{z}^{(i)}(t)}}{\sum_{k=1}^{m} e^{-\mathbf{z}^{(k)}(t)}}. \tag{10}$$

Note that SM-NLMS algorithm also could be constructed through gradient descent method with stochastic cost function defined as

$$F(e(t)) \triangleq \begin{cases} \left(\frac{|e(t)|-\gamma}{\|\mathbf{y}(t)\|}\right)^2 & |e(t)| > \gamma \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, for the unconstrained parameter vector update, stochastic gradient algorithm is given by

$$\mathbf{z}(t+1) = \mathbf{z}(t) - \frac{1}{2}\nabla_{\mathbf{z}} F(e(t)) \tag{11}$$

which by chain rule yields to

$$\mathbf{z}(t+1) = \mathbf{z}(t) - \frac{1}{2}[\nabla_{\mathbf{z}}\mathbf{w}(t)]^T\nabla_{\mathbf{w}} F(e(t)). \tag{12}$$

Note that $\nabla_{\mathbf{z}}\mathbf{w}(t) = \mathbf{w}(t)\mathbf{w}(t)^T - \text{diag}\{\mathbf{w}(t)\}$ [8] and by this we obtain

$$\mathbf{z}(t+1) = \mathbf{z}(t) + \mu(t)[\mathbf{w}(t)\mathbf{w}(t)^T - \text{diag}\{\mathbf{w}(t)\}]\frac{\mathbf{y}(t)e(t)}{\mathbf{y}(t)^T\mathbf{y}(t)} \tag{13}$$

where

$$\mu(t) = \begin{cases} 1 - \frac{\gamma}{|e(t)|} & \text{if } |e(t)| > \gamma, \\ 0 & \text{otherwise.} \end{cases}$$

and

$$\mathbf{w}(t) = \frac{e^{-\mathbf{z}(t)}}{\|e^{-\mathbf{z}(t)}\|_1}.$$

With the algorithms defined above, in next section we evaluate the MSE performance of the algorithms within different schemes.

5. SIMULATIONS AND RESULTS

In this section, through wide range of benchmark real life datasets and simulations, we demonstrate the performance of the proposed SMF filter mixture algorithms and compare the steady-state and convergence performances with various methods, i.e. NLMS, variable step size NLMS (VSS) and affine projection algorithm (APA), as well as its superior computational efficiency [13, 14].

5.1. Stationary Data

In this part, we study our algorithms in a stationary environment where data source statistics do not change over time. We use input vectors with eigenvalue spread of 1 and 0 dB SNR signal, where σ_n^2 represents the white Gaussian noise variance. Parameter of interest chosen randomly from normal distribution and normalized to $||\mathbf{w}_o|| = 1$. We use 10 constituent SM-NLMS filters with different error bounds set around $\sqrt{5\sigma_n^2}$. For comparison, we used NLMS mixture algorithm and a single NLMS algorithm with step size $\mu_{NLMS} = 0.2$, VSS-NLMS algorithm with step size range $(\mu_{max}, \mu_{min}) = (0.2, 0.02)$ and APA algorithm of order 5. In Fig.2, we demonstrate the time accumulated regression errors averaged over 100 independent trials. We observe that SMF and NLMS mixture of set membership filters outperform other filters (NLMS, VSS-NLMS and APA) in both convergence rate and residual error sense. Also, note that SMF mixture algorithms achieve better steady state error than the NLMS mixture algorithm.

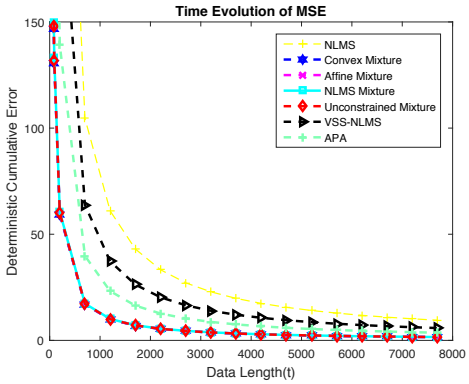

Fig. 2: Time accumulated error performance of proposed algorithms compared with other algorithms over stationary data having 0dB SNR and input vector eigenvalue spread of 1.

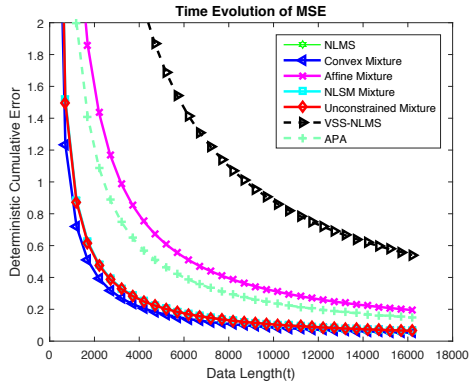

Fig. 3: Time accumulated error performance of proposed algorithms compared with NLMS algorithms over Pumadyn and Elevator datasets.

5.2. Benchmark Real Data

Finally, we apply our algorithms to the regression of the benchmark real-life problems [15]. In real-life dataset experiments, we use 10 constituent SMF filters and since this time we do not know the power of the additive noise, we set the error bounds of the SMF filters in a wide range spread around 0.15 and again we choose the error bound for the combinator SMF filter as 0.15. We make 100 trials over a dataset by shuffling the data at each trial.

We use Elevator data with regressor dimension 18 which is a dataset obtained from the task of controlling F16 aircraft and the desired data is related to an action taken on the elevators of the aircraft [15]. We set the order of APA algorithm as 8 for this case. We present the results for this dataset in Fig.3. Note that in Fig.3, mixture approaches show superior performance over other filters. Although APA algorithm shows a close performance to mixture filters, we emphasize that APA algorithm is computationally inefficient for big data applications compared to proposed methods since it requires memory for holding old data at its order and require more multiplication and addition operations at each update. We present detailed results for that in the computational load analysis part.

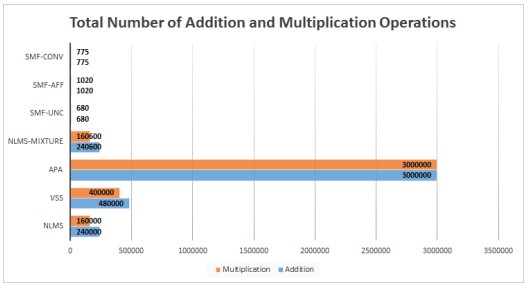

Fig. 4: Number of operations that each algorithm requires over 8000 instance stationary data.

5.3. Computational Load

One of the critical aspects of the proposed algorithms is the reduced computational load regarding lessened update of weights compared to the standard NLMS algorithms and mixture methods. To present that, we calculated the total number of addition and multiplication operation that each algorithm made during the simulation. In Fig.4, we demonstrate results for addition and multiplication operation that each algorithm made in 100 independent experiment over stationary data and show that proposed algorithms computationally more efficient than other algorithms. Although, the computational cost among the proposed algorithms do not differ much, we emphasize that the unconstrained mixture is the most computationally efficient one. We note that SMF mixture algorithms provide computational efficiency up to order of magnitude of 3.

6. CONCLUSION

In this paper we introduced a novel mixture of expert algorithm in order to reduce the computational demand of the mixture approaches. Since the ordinary mixture approaches require to run several adaptive filters in parallel, they are impractical in applications involving big data due to complexity issues. To this end, by using the SMF, we signficantly reduced the computational complexity of these approaches while providing superior performance. We provided unconstrained, affine and convex mixture weight configurations using set membership filtering framework. Through numerical experiments in stationary environments and through regression of a bencmark real life problem, we investigated the steady-state mean square error and convergence rate performance of these algorithms compared with other algorithms and mixture methods. In these experiments we demonstrated that proposed algorithms reach faster convergence rate and lower steady state error. Finally, we showed that our set membership filtering based approaches requires less addition and multiplication operations hence less computational load than the compared algorithms.

7. REFERENCES

[1] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1078–1090, 2006.

[2] J. Arenas-Garcia, V. Gomez-Verdejo, and A. R. Figueiras-Vidal, "New algorithms for improved adaptive convex combination of LMS transversal filters," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 6, pp. 2239–2249, 2005.

[3] J. Arenas-Garcia, M. Martinez-Ramon, V. Gomez-Verdejo, and A. R. Figueiras-Vidal, "Multiple plant identifier via adaptive LMS convex combination," in *2003 IEEE International Symposium on Intelligent Signal Processing*, 2003, pp. 137–142.

[4] V. H. Nascimento, M. T. M. Silva, and J. Arenas-Garcia, "A low-cost implementation strategy for combinations of adaptive filters," in *Proc. Int. Conf. Acoust., Speech, and Signal Process. (ICASSP)*, 2013, pp. 5671–5675.

[5] Jing Lu, Steven Hoi, and Peilin Zhao, "Second order online collaborative filtering," in *Proceedings of Asian Conference on Machine Learning (ACML)*, 2013, pp. 40–55.

[6] Guang Ling and Haiqin Yang, "Online learning for collaborative filtering," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, 2012, pp. 1–8.

[7] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y. F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Processing Letters*, vol. 5, no. 5, 1998.

[8] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, "Steady-state MSE performance analysis of mixture approaches to adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 58, pp. 4050–4063, 2010.

[9] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, "A transient analysis of adaptive affine combinations," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 6227–6232, 2011.

[10] M. A. Donmez and S. S. Kozat, "Steady-state MSE analysis of convexly constrained mixture methods," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 3314–3321, 2012.

[11] P. S. R. Diniz and S. Werner, "Set-membership binormalized data-reusing LMS algorithms," *IEEE Transactions on Signal Processing*, vol. 51, no. 1, pp. 124–134, 2003.

[12] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer, "Online passive-aggressive algorithms," *The Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

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

[14] H. Zhao and Y. Yu, "Novel adaptive vss-nlms algorithm for system identification," in *Intelligent Control and Information Processing (ICICIP), 2013 Fourth International Conference on*, June 2013, pp. 760–764.

[15] J. Alcala-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garca, L. Snchez, and F. Herrera, "Keel data-mining software tool: Data set repository,integration of algorithms and experimental analysis framework," .