Improved Data-Selective LMS-Newton Adaptation Algorithms

Md. Zulfiquar A. Bhotto and Andreas Antoniou

Dept. of Elec. and Comp. Eng.
University of Victoria
Victoria, BC, Canada
Email: zbhotto@ece.uvic.ca
aantoniou@ieee.org

16th International Conference on Digital Signal Processing
July 5–7, 2009

University of Victoria

Outline

- Known LMS-Newton adaptation algorithms

Outline

- Known LMS-Newton adaptation algorithms
- Proposed data-selective LMS-Newton adaptation algorithms

University of Victoria

Outline

- Known LMS-Newton adaptation algorithms
- Proposed data-selective LMS-Newton adaptation algorithms
- Simulation results and comparisons

Outline

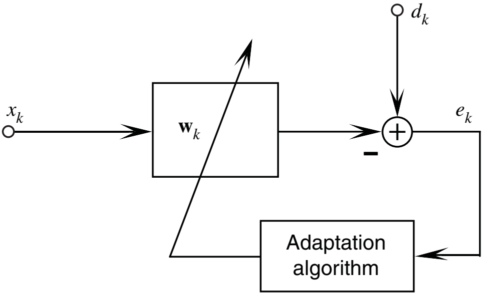
- Known LMS-Newton adaptation algorithms
- Proposed data-selective LMS-Newton adaptation algorithms
- Simulation results and comparisons
- Conclusions

Adaptation Algorithms

- Usually adaptation algorithms use the *a posteriori* error, ε_k, at iteration k given by

$$e_k = d_k - \mathbf{w}_k^T \mathbf{x}_k$$

to adjust the weight vector $\mathbf{w}_k$ using the input signal vector $\mathbf{x}_k$ and desired signal d_k.



Known LMS-Newton Adaptation Algorithms

- The basic LMS-Newton algorithm (Farhang-Boroujeny and Gazor, *IEE Proc.*, 1991) solves the optimization problem

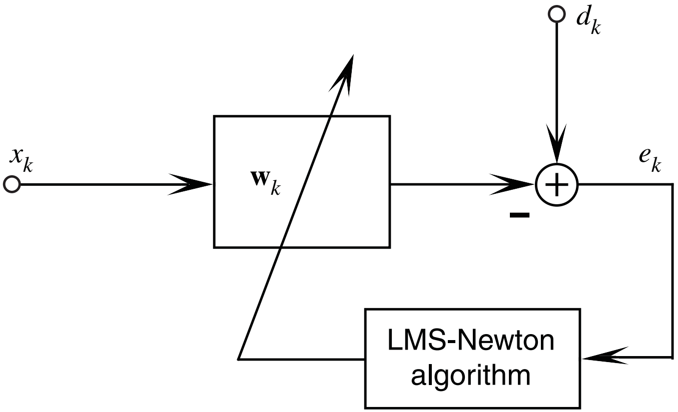
$$\underset{\mathbf{w}_k}{\text{minimize}} \quad E\left[(d_k - \mathbf{w}_k^T \mathbf{x}_k)^2\right]$$

by using the update equations

$$\begin{aligned} z_k &= \frac{1-\alpha}{\alpha} + \mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k \\ \mathbf{w}_k &= \mathbf{w}_{k-1} + \frac{2\mu}{\alpha} \frac{e_k \hat{R}_{k-1}^{-1} \mathbf{x}_k}{z_k} \\ \hat{R}_k^{-1} &= \frac{1}{1-\alpha} \left(\hat{R}_{k-1}^{-1} - \frac{\hat{R}_{k-1}^{-1} \mathbf{x}_k \mathbf{x}_k^T \hat{R}_{k-1}^{-1}}{z_k} \right) \end{aligned}$$

where $e_k = d_k - \mathbf{w}_{k-1}^T \mathbf{x}_k$ is the *a priori* error at iteration k, $\hat{R}_k^{-1}$ is an estimate of the inverse of the input-signal autocorrelation matrix, μ is the *step size*, and α is the *convergence factor*.

Known LMS-Newton Adaptation Algorithms *Cont'd*



Known LMS-Newton Adaptation Algorithms

- Two specific LMS-Newton adaptation algorithms, referred to as *Algorithms I and II*, were described by Diniz, de Campos, and Antoniou in *IEEE Transactions on Signal Processing* in 1995.

Algorithm I of Diniz et al.

- Algorithm I uses a *variable convergence factor*

$$\alpha_k = \frac{1}{1 + (2b - 1)\mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k}$$

and *fixed step size* $\mu_k = b\alpha_k$ where $b > 0.5$ in the basic LMS-Newton algorithm.

Algorithm I of Diniz et al.

- Algorithm I uses a *variable convergence factor*

$$\alpha_k = \frac{1}{1 + (2b - 1)\mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k}$$

and *fixed step size* $\mu_k = b\alpha_k$ where $b > 0.5$ in the basic LMS-Newton algorithm.

- The update equations are:

$$\begin{aligned} z_k &= 2b\mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k \\ \mathbf{w}_k &= \mathbf{w}_{k-1} + 2\mu_k e_k \hat{R}_{k-1}^{-1} \mathbf{x}_k \\ \hat{R}_k^{-1} &= \frac{1 + (1 - 0.5/b) z_k}{(1 - 0.5/b) z_k} \left(\hat{R}_{k-1}^{-1} - \frac{\hat{R}_{k-1}^{-1} \mathbf{x}_k \mathbf{x}_k^T \hat{R}_{k-1}^{-1}}{z_k} \right) \end{aligned}$$

Algorithm II of Diniz et al.

- Algorithm II uses a *variable step size* μ_k

$$\mu_k = \frac{1}{2\mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k}$$

and a *fixed convergence factor* α in the basic LMS-Newton algorithm.

Algorithm II of Diniz et al.

- Algorithm II uses a *variable step size* μ_k

$$\mu_k = \frac{1}{2\mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k}$$

and a *fixed convergence factor* α in the basic LMS-Newton algorithm.

- The update equations assume the form:

$$\begin{aligned}
z_k &= \frac{1-\alpha}{\alpha} + \mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k \\
\mathbf{w}_k &= \mathbf{w}_{k-1} + 2\mu_k e_k \hat{R}_{k-1}^{-1} \mathbf{x}_k \\
\hat{R}_k^{-1} &= \frac{1}{1-\alpha} \left(\hat{R}_{k-1}^{-1} - \frac{\hat{R}_{k-1}^{-1} \mathbf{x}_k \mathbf{x}_k^T \hat{R}_{k-1}^{-1}}{z_k} \right)
\end{aligned}$$

University of Victoria

Algorithm I Versus Algorithm II

- Algorithm I is preferred if the input signal statistics (mean, variance) are not known a priori.

Algorithm I Versus Algorithm II

- Algorithm I is preferred if the input signal statistics (mean, variance) are not known a priori.
- Otherwise, Algorithm II is preferred.

Algorithm I Versus Algorithm II

- Algorithm I is preferred if the input signal statistics (mean, variance) are not known a priori.

- Otherwise, Algorithm II is preferred.

- In Algorithms I and II, a reduction factor, q, can be introduced by modifying the update equation as

$$\mathbf{w}_k = \mathbf{w}_{k-1} + 2q\mu_k e_k \hat{R}_{k-1}^{-1}\mathbf{x}_k$$

By using $q = 1$, *fast convergence* can be achieved.

Algorithm I Versus Algorithm II

- Algorithm I is preferred if the input signal statistics (mean, variance) are not known a priori.
- Otherwise, Algorithm II is preferred.
- In Algorithms I and II, a reduction factor, q, can be introduced by modifying the update equation as

$$\mathbf{w}_k = \mathbf{w}_{k-1} + 2q\mu_k e_k \hat{R}_{k-1}^{-1}\mathbf{x}_k$$

By using $q = 1$, *fast convergence* can be achieved.

On the other hand, by using a suitable value of q less than one, *minimum misalignment* can be achieved.

Modified LMS-Newton Adaptation Algorithms I and II

- The proposed two LMS-Newton algorithms are essentially *modified versions* of Algorithms I and II reported by Diniz et al.

Modified LMS-Newton Adaptation Algorithms I and II

- The proposed two LMS-Newton algorithms are essentially *modified versions* of Algorithms I and II reported by Diniz et al.

- In these algorithms, we use the step size that solves the optimization problem

$$\mu_k = \begin{cases} \underset{\mu_k}{\text{argmin}} \quad \left(|d_k - \mathbf{x}_k^T \mathbf{w}_k| - \gamma\right) & \text{if } |e_k| > \gamma \\ 0 & \text{otherwise} \end{cases}$$

where e_k is the *a priori* error at iteration k and γ is a prespecified error bound.

Modified LMS-Newton Adaptation Algorithms I and II

- The proposed two LMS-Newton algorithms are essentially *modified versions* of Algorithms I and II reported by Diniz et al.

- In these algorithms, we use the step size that solves the optimization problem

$$\mu_k = \begin{cases} \underset{\mu_k}{\text{argmin}} \quad \left(|d_k - \mathbf{x}_k^T \mathbf{w}_k| - \gamma\right) & \text{if } |e_k| > \gamma \\ 0 & \text{otherwise} \end{cases}$$

where e_k is the *a priori* error at iteration k and γ is a prespecified error bound.

- The required μ_k can be deduced as $\mu_k = \frac{\beta_k}{2\mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k}$

where
$$\beta_k = \begin{cases} 1 - \dfrac{\gamma}{|e_k|} & \text{if } |e_k| > \gamma \\ 0 & \text{otherwise} \end{cases}$$

Modified Algorithms I and II *Cont'd*

- The step size μ_k forces the equality $|e_k| = \gamma$ whenever the magnitude of the *a priori* error at iteration k assumes a value greater than γ.

Modified Algorithms I and II *Cont'd*

- The step size μ_k forces the equality $|e_k| = \gamma$ whenever the magnitude of the *a priori* error at iteration k assumes a value greater than γ.
- The update equations of improved Algorithm I are:

$$
\begin{aligned}
z_k &= 2b\mathbf{x}_k^T \hat{R}_{k-1}^{-1}\mathbf{x}_k \\
\mathbf{w}_k &= \mathbf{w}_{k-1} + 2\mu_k e_k \hat{R}_{k-1}^{-1}\mathbf{x}_k \text{ with } \mu_k = \frac{\beta_k}{2\mathbf{x}_k^T \hat{R}_{k-1}^{-1}\mathbf{x}_k} \\
\hat{R}_k^{-1} &= \frac{1 + (1-0.5/b)z_k}{(1-0.5/b)z_k}\left(\hat{R}_{k-1}^{-1} - \frac{\hat{R}_{k-1}^{-1}\mathbf{x}_k\mathbf{x}_k^T\hat{R}_{k-1}^{-1}}{z_k}\right)
\end{aligned}
$$

University of Victoria

Modified Algorithms I and II *Cont'd*

- The update equations of improved LMS-Newton Algorithm II are:

$$z_k = \frac{1-\alpha}{\alpha} + \mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k$$

$$\mathbf{w}_k = \mathbf{w}_{k-1} + 2\mu_k e_k \hat{R}_{k-1}^{-1} \mathbf{x}_k \text{ with } \mu_k = \frac{\beta_k}{2\mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k}$$

$$\hat{R}_k^{-1} = \frac{1}{1-\alpha} \left(\hat{R}_{k-1}^{-1} - \frac{\hat{R}_{k-1}^{-1} \mathbf{x}_k \mathbf{x}_k^T \hat{R}_{k-1}^{-1}}{z_k} \right)$$

Modified Algorithms I and II *Cont'd*

- The update equations of improved LMS-Newton Algorithm II are:

$$
\begin{aligned}
z_k &= \frac{1-\alpha}{\alpha} + \mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k \\
\mathbf{w}_k &= \mathbf{w}_{k-1} + 2\mu_k e_k \hat{R}_{k-1}^{-1} \mathbf{x}_k \text{ with } \mu_k = \frac{\beta_k}{2\mathbf{x}_k^T \hat{R}_{k-1}^{-1} \mathbf{x}_k} \\
\hat{R}_k^{-1} &= \frac{1}{1-\alpha} \left(\hat{R}_{k-1}^{-1} - \frac{\hat{R}_{k-1}^{-1} \mathbf{x}_k \mathbf{x}_k^T \hat{R}_{k-1}^{-1}}{z_k} \right)
\end{aligned}
$$

- Since $0 \leq \beta_k < 1$, β_k acts as a *variable reduction factor* and, therefore, a reduced *steady-state misalignment* would be obtained without reducing the *convergence speed*.

- The reduction factor β_k tends to remain close to unity during transience and hence the *convergence speed* of the improved algorithm tends to be similar to that of the known algorithm.

Modified Algorithms I and II *Cont'd*

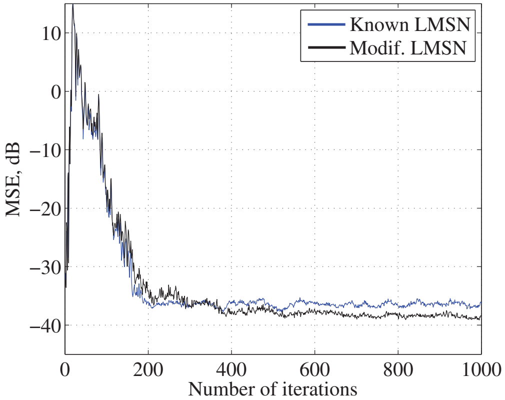
- The reduction factor β_k tends to remain close to unity during transience and hence the *convergence speed* of the improved algorithm tends to be similar to that of the known algorithm.
- At steady state, the step size β_k approaches zero and, consequently, a *reduced steady-state misalignment* is achieved.

Modified Algorithms I and II *Cont'd*

- The reduction factor β_k tends to remain close to unity during transience and hence the *convergence speed* of the improved algorithm tends to be similar to that of the known algorithm.
- At steady state, the step size β_k approaches zero and, consequently, a *reduced steady-state misalignment* is achieved.
- Since an update is performed *only if* the threshold the *a priori* error exceeds threshold γ, a *significant reduction in the number updates*, and hence in the amount of computation, is achieved.

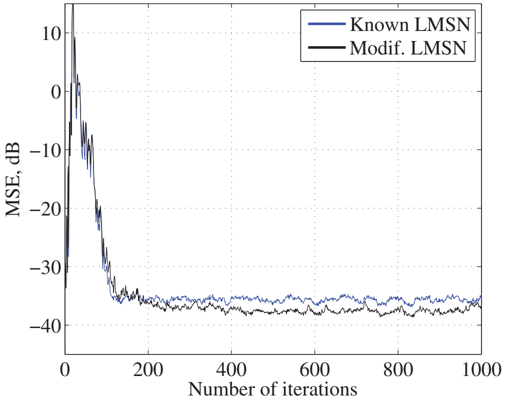
Simulation Results – Algorithm I

- Learning curves for a system identification problem in a stationary environment:



Simulation Results – Algorithm I *Cont'd*

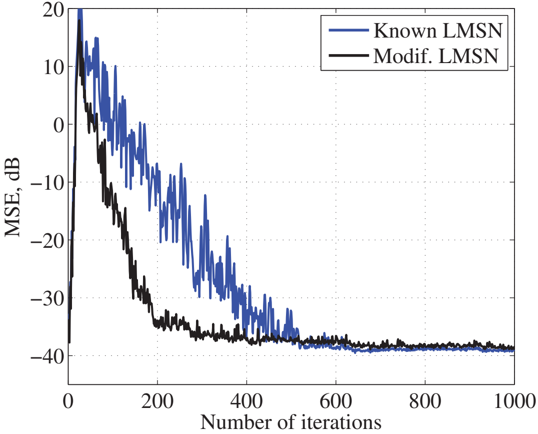
- Learning curves for a system identification problem in a nonstationary environment:



University of Victoria

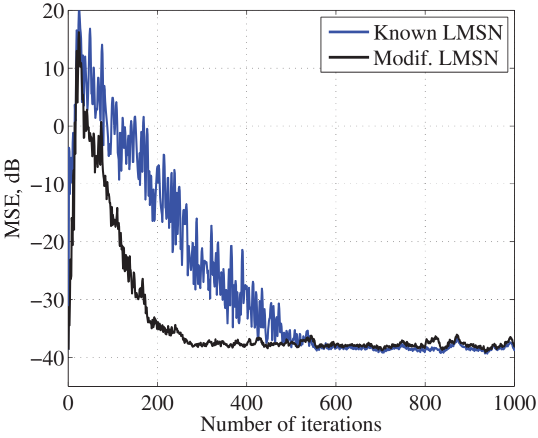
Simulation Results – Algorithm I *Cont'd*

- Learning curves for a system identification problem in a stationary environment (reduction factor in known algorithm $q = 0.34$):



Simulation Results – Algorithm I *Cont'd*

- Learning curves for a system identification problem in a nonstationary environment (reduction factor in known algorithm $q = 0.34$):



Simulation Results – Algorithm I *Cont'd*

- Steady-state misalignment in a stationary environment:

SNR	Algorithm I	MSE in dB with data length, N		
		1000	5000	10000
20 dB	Known	−16.50	−16.48	−16.44
	Modified	−18.60	−19.20	−19.26
	Difference	2.10	2.72	2.82
30 dB	Known	−26.54	−26.46	−26.44
	Modified	−28.54	−29.20	−29.10
	Difference	2.00	2.74	2.66
40 dB	Known	−36.51	−36.54	−36.51
	Modified	−38.61	−39.17	−39.20
	Difference	2.10	2.63	2.69

Simulation Results – Algorithm I *Cont'd*

- Steady-state misalignment in nonstationary environment:

SNR	Algorithm I	MSE in dB with data length, N		
		1000	5000	10000
20 dB	Known	−16.55	−16.64	−16.58
	Modified	−18.71	−18.67	−18.50
	Difference	2.16	2.03	1.92
30 dB	Known	−26.39	−26.47	−26.49
	Modified	−28.65	−28.55	−28.62
	Difference	2.26	2.08	2.13
40 dB	Known	−36.49	−36.40	−36.55
	Modified	−38.63	−38.58	−38.59
	Difference	2.14	2.18	2.04

Simulation Results – Algorithm II

- Similar simulation results to those presented have been obtained for modified Algorithm II and are presented in the paper.

Simulation Results – Number of Updates *Cont'd*

- Updates required in 1000 iterations:

Exp.	Algorithm	Weight updates	Reduction, %
1	I	210	79
2	II	190	81
3	I	309	69
4	II	274	72
5	I	222	77
6	II	198	80

Conclusions

- Improved versions of the LMSN algorithms proposed by Diniz et al. have been proposed.

University of Victoria

Conclusions

- Improved versions of the LMSN algorithms proposed by Diniz et al. have been proposed.
- They yield a reduced *steady-state misalignment* relative to that in the known LMSN algorithms while requiring a *similar number of iterations to converge* in stationary as well as nonstationary environments.

Conclusions

- Improved versions of the LMSN algorithms proposed by Diniz et al. have been proposed.
- They yield a reduced *steady-state misalignment* relative to that in the known LMSN algorithms while requiring a *similar number of iterations to converge* in stationary as well as nonstationary environments.
- Using a reduction factor $q = 0.34$ in the known algorithms, the modified algorithms require *a reduced number of iterations* to converged while achieving approximately the *same misalignment* as the known algorithms.

Conclusions

- Improved versions of the LMSN algorithms proposed by Diniz et al. have been proposed.
- They yield a reduced *steady-state misalignment* relative to that in the known LMSN algorithms while requiring a *similar number of iterations to converge* in stationary as well as nonstationary environments.
- Using a reduction factor $q = 0.34$ in the known algorithms, the modified algorithms require *a reduced number of iterations* to converged while achieving approximately the *same misalignment* as the known algorithms.
- The modified algorithms require a reduced number of updates of the order of 70% or more, which would lead to a *significant reduction in the computational effort*.

University of Victoria

Thank you for your attention.
Any questions?