4F7 Adaptive Filters (and Spectrum Estimation)

Least Mean Square (LMS) Algorithm

Sumeetpal Singh
Engineering Department
Email : sss40@eng.cam.ac.uk

1 Outline

- The LMS algorithm
- Overview of LMS issues concerning step-size bound and convergence
- Some simulation examples
- The normalised LMS (NLMS)

2 Least Mean Square (LMS)

- Steepest Descent (SD) was

$$\begin{aligned}\mathbf{h}(n+1) &= \mathbf{h}(n) - \frac{\mu}{2}\nabla J(\mathbf{h}(n)) \\ &= \mathbf{h}(n) + \mu E\{\mathbf{u}(n)e(n)\}\end{aligned}$$

- Often $\nabla J(\mathbf{h}(n)) = -2E\{\mathbf{u}(n)e(n)\}$ is unknown or too difficult to derive
- Remedy is to use the instantaneous approximation $-2\mathbf{u}(n)e(n)$ for $\nabla J(\mathbf{h}(n))$
- Using this approximation we get the LMS algorithm

$$\begin{aligned}e(n) &= d(n) - \mathbf{h}^{\mathsf{T}}(n)\mathbf{u}(n), \\ \mathbf{h}(n+1) &= \mathbf{h}(n) + \mu e(n)\mathbf{u}(n)\end{aligned}$$

- This is desirable because
  - We do not need knowledge of $\mathbf{R}$ and $\mathbf{p}$ anymore
  - If statistics are changing over time, it adapts accordingly
  - Complexity: $2M+1$ multiplications and $2M$ additions per iteration. Not M^2 multiplications like SD
- Undesirable because we have to choose μ when $\mathbf{R}$ not known, subtle convergence analysis

3 Application: Noise Cancellation

- Mic 1: Reference signal

$$d(n) = \underbrace{s(n)}_{\text{signal of interest}} + \underbrace{v(n)}_{\text{noise}}$$

  $s(n)$ and $v(n)$ statistically independent

- Aim: recover signal of interest
- Method: use another mic, Mic 2, to record noise only, $u(n)$
- Although $u(n) \neq v(n)$, $u(n)$ and $v(n)$ are correlated
- Now **filter** recorded noise $u(n)$ to minimise $E\{e(n)^2\}$, i.e. to cancel $v(n)$
- Recovered signal is $e(n) = d(n) - \mathbf{h}(n)^T \mathbf{u}(n)$ and not $y(n)$
- **Run** Matlab demo on webpage

- We are going to see an example with speech $s(n)$ generated as a mean 0 variance 1 Gaussian random variable
- Mic 1's noise was $0.5\sin(n\frac{\pi}{2}+0.5)$
- Mic 2's noise was $10\sin(n\frac{\pi}{2})$
- Mic 1 and 2's noise are both sinusoids but with different amplitudes and phase shifts
- You could increase the phase shift but you will need a larger value for M
- **Run** Matlab demo on webpage

4 LMS convergence in mean

- Write the reference signal model as

$$d(n) = \mathbf{u}^{\mathtt{T}}(n)\,\mathbf{h}_{\text{opt}} + \varepsilon(n)$$
$$\varepsilon(n) = d(n) - \mathbf{u}^{\mathtt{T}}(n)\,\mathbf{h}_{\text{opt}}$$

where $\mathbf{h}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{p}$ denotes the optimal vector (Wiener filter) that $\mathbf{h}(n)$ should converge to

- For this reference signal model, the LMS becomes

$$\begin{aligned}
\mathbf{h}(n+1) &= \mathbf{h}(n) + \mu\mathbf{u}(n) \\
&\times \left(\mathbf{u}^{\mathtt{T}}(n)\,\mathbf{h}_{\text{opt}} + \varepsilon(n) - \mathbf{u}^{\mathtt{T}}(n)\,\mathbf{h}(n)\right) \\
&= \mathbf{h}(n) + \mu\mathbf{u}(n)\,\mathbf{u}^{\mathtt{T}}(n) \\
&\times \left(\mathbf{h}_{\text{opt}} - \mathbf{h}(n)\right) + \mu\mathbf{u}(n)\,\varepsilon(n) \\
\mathbf{h}(n+1) - \mathbf{h}_{\text{opt}} &= \left(\mathbf{I} - \mu\mathbf{u}(n)\,\mathbf{u}^{\mathtt{T}}(n)\right)\left(\mathbf{h}(n) - \mathbf{h}_{\text{opt}}\right) \\
&+ \mu\mathbf{u}(n)\,\varepsilon(n)
\end{aligned}$$

- This looks like a noisy version of the SD recursion

$$\mathbf{h}(n+1) - \mathbf{h}_{\text{opt}} = (\mathbf{I} - \mu\mathbf{R})\left(\mathbf{h}(n) - \mathbf{h}_{\text{opt}}\right)$$

- Verify that $E\{\mathbf{u}(n)\varepsilon(n)\} = 0$ using $\mathbf{h}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{p}$
- Introducing the expectation operator gives

$$\begin{aligned}
&E\left\{\mathbf{h}(n+1) - \mathbf{h}_{\text{opt}}\right\} \\
&= E\left\{\left(\mathbf{I} - \mu\mathbf{u}(n)\mathbf{u}^{\mathtt{T}}(n)\right)\left(\mathbf{h}(n) - \mathbf{h}_{\text{opt}}\right)\right\} \\
&+ \mu\underbrace{E\left\{\mathbf{u}(n)\varepsilon(n)\right\}}_{=0} \\
&\approx \left(\mathbf{I} - \mu E\left\{\mathbf{u}(n)\mathbf{u}^{\mathtt{T}}(n)\right\}\right) E\left\{\mathbf{h}(n) - \mathbf{h}_{\text{opt}}\right\} \\
&\quad \text{(Independence approximation)} \\
&= (\mathbf{I} - \mu\mathbf{R}) E\left\{\mathbf{h}(n) - \mathbf{h}_{\text{opt}}\right\}
\end{aligned}$$

- *Independence approximation* assumes $\mathbf{h}(n) - \mathbf{h}_{\mathrm{opt}}$ is independent of $\mathbf{u}(n)\,\mathbf{u}^{\mathsf{T}}(n)$
  – Since $\mathbf{h}(n)$ function of $\mathbf{u}(0)$, $\mathbf{u}(1)$, …, $\mathbf{u}(n-1)$ and all previous desired signals this is not true
  – However, the approximation is better justified for a “block” LMS type update scheme where the filter is updated at multiples of some block length L, i.e. when $n = kL$ and not otherwise

- Idea is to not update $\mathbf{h}(n)$ except when n is an integer multiple of L, i.e. $n = kL$ for $k = 0, 1, \ldots$

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu(n+1)e(n)\mathbf{u}(n)$$
$$e(n) = d(n) - \mathbf{h}(n)^T\mathbf{u}(n)$$
$$\mu(n) = \begin{cases} \mu & \text{if } n/L = \text{integer} \\ 0 & \text{otherwise} \end{cases}$$

- Also L should be much larger than filter length M
- This means $\mathbf{h}(kL) = \mathbf{h}(kL+1) = \cdots = \mathbf{h}(kL+L-1)$
- Re-use the previous derivation which is still valid:

$$E\left\{\mathbf{h}(n+1) - \mathbf{h}_{\text{opt}}\right\} = E\left\{\left(\mathbf{I} - \mu(n+1)\mathbf{u}(n)\mathbf{u}(n)^T\right)\left(\mathbf{h}(n) - \mathbf{h}_{\text{opt}}\right)\right\}$$

- When $n+1 = kL + L$ we have

$$E\left\{\mathbf{h}(n+1) - \mathbf{h}_{\text{opt}}\right\} = E\left\{\left(\mathbf{I} - \mu\mathbf{u}(n)\mathbf{u}(n)^T\right)\left(\mathbf{h}(kL) - \mathbf{h}_{\text{opt}}\right)\right\}$$
$$\approx E\left\{\mathbf{I} - \mu\mathbf{u}(n)\mathbf{u}(n)^T\right\} E\left\{\mathbf{h}(kL) - \mathbf{h}_{\text{opt}}\right\}$$
$$E\left\{\mathbf{h}(kL+L) - \mathbf{h}_{\text{opt}}\right\} \approx (\mathbf{I} - \mu\mathbf{R})\, E\left\{\mathbf{h}(kL) - \mathbf{h}_{\text{opt}}\right\}$$

- This analysis uses the fact that $(u(0), \ldots, u(i))$ and $(u(j), \ldots, u(j+M-1))$, for $j > i$, become independent as $j - i$ increases. True for some ARMA time-series.

- We are back to the SD scenario and so

$$E\left\{\mathbf{h}\,(n)\right\} \to \mathbf{h}_{\text{opt}} \quad \text{if } 0 < \mu < \frac{2}{\lambda_{\max}}$$

- Behaviour predicted using the analysis of the block LMS agrees with experiments and computer simulations even for $L = 1$

- We will always use $\mu(n) = \mu$ for all n. Block LMS version just to understand long-term behaviour

- The point of the LMS was that we don't have access to $\mathbf{R}$, so how to compute $\lambda_{\max}$?
- Using the fact that

$$\sum_{k=1}^{M} \lambda_k = \mathrm{tr}\left(\mathbf{R}\right) = ME\left\{u^2\left(n\right)\right\}$$

we have that $\lambda_{\max} < \sum_{k=1}^{M} \lambda_k = ME\left\{u^2\left(n\right)\right\}$

- Note that we can estimate $E\left\{u^2\left(n\right)\right\}$ by a simple sample average and the new tighter bound on the stepsize is

$$0 < \mu < \frac{2}{ME\left\{u^2\left(n\right)\right\}} < \frac{2}{\lambda_{\max}}$$

- With a fixed stepsize, $\{\mathbf{h}(n)\}_{n\geq 0}$ will never settle at $\mathbf{h}_{\text{opt}}$, but rather oscillate about $\mathbf{h}_{\text{opt}}$. Even if $\mathbf{h}(n) = \mathbf{h}_{\text{opt}}$ then

$$\mathbf{h}(n+1) - \mathbf{h}_{\text{opt}} = \mu\mathbf{u}(n)e(n) = \mu\mathbf{u}(n)\left(d(n) - \mathbf{u}^{\mathtt{T}}(n)\,\mathbf{h}_{\text{opt}}\right),$$

and because $\mathbf{u}(n)\,e(n)$ is random, $\mathbf{h}(n+1)$ will move away from $\mathbf{h}_{\text{opt}}$

5 LMS main points

- Simple to implement
- Works fine is many applications if filter order and stepsize is chosen properly
- There is a trade-off effect with the stepsize choice
  – large μ yields better tracking ability in a non-stationary environment but will have larger fluctuations of $\mathbf{h}(n)$ about converged value
  – small μ has poorer tracking ability but less of such fluctuations

6 Adaptive stepsize: Normalised LMS (NLMS)

- We showed that LMS was stable provided
$$\mu < \frac{2}{ME\left\{u^2(n)\right\}}$$
- What if $E\left\{u^2(n)\right\}$ varied, which would be true for a non-stationary input signal
- LMS should be able to adapt its step-size automatically
- The instantaneous estimate of $ME\left\{u^2(n)\right\}$ is $\mathbf{u}^{\mathrm{T}}(n)\,\mathbf{u}(n)$
- Now replace the LMS stepsize with $\frac{\mu'}{\mathbf{u}^{\mathrm{T}}(n)\mathbf{u}(n)} = \frac{\mu'}{\|\mathbf{u}(n)\|^2}$ where μ' is a constant that should < 2, e.g. , $0.25 < \mu' < 0.75$. We make μ' smaller because of the poor quality estimate for $ME\left\{u^2(n)\right\}$ in the denominator

- This choice of stepsize gives the **Normalized Least Mean Squares** (NLMS)

$$e(n) = d(n) - \mathbf{u}^{\mathtt{T}}(n)\,\mathbf{h}(n)$$
$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2} e(n)\,\mathbf{u}(n)$$

  where μ' is relabelled to μ. NLMS is the LMS algorithm with a **data-dependent stepsize**

- Note small amplitudes will now adversely effect the NLMS. To better stabilise the NLMS use

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2 + \epsilon} e(n)\,\mathbf{u}(n)$$

  where ϵ is a small constant, e.g. 0.0001.

7 Comparing NLMS and LMS

- Compare the stability of the LMS and NLMS for different values of stepsize. You will see that the NLMS is stable for $0 < \mu < 2$. You will still need to tune μ to get the desired convergence behaviour (or fluctuations of $\mathbf{h}(n)$ once it has stabilized) though.
- **Run** the NLMS example on the course website