An Analysis of Channel Estimation and Equalization Techniques Based on Affine Projection Algorithm

Yanan Xiao

Northwestern Polytechnic University

Graduation Design Thesis Presentation

Outline

Study in Noisy Environment

Judge how noisy the channel is—suitable for you to learn, or too noisy.

Tweet under Supervision

Judge whether it is safe to post something on Weibo.com.

Performance (Cost) Functions

- Mean Square Error $E[e^2(n)]$ (Most popular)
  Adaptive algorithms: Least Mean Square (LMS), Normalized LMS (NLMS), Affine Projection (AP), Recursive Least Squares (RLS), etc.
- Regularized MSE

$$J_{rms} = E[e^2(n)] + \alpha \|\mathbf{w}(n)\|^2 \quad (1)$$

  Adaptive algorithm: Leaky Least Mean Square (leaky LMS)
- l_1 norm criterion

$$J_{l_1} = E[|e(n)|] \quad (2)$$

  Adaptive algorithm: Sign-Error

- Least Mean Fourth (LMF) criterion

$$J_{LMF} = E[e^4(n)] \tag{3}$$

Adaptive algorithm: Least Mean Fourth (LMF)

- Least Mean Mixed Norm (LMMN) criterion

$$J_{LMMN} = E[\alpha e^2(n) + \frac{1}{2}(1-\alpha)e^4(n)] \tag{4}$$

Adaptive algorithm: Least Mean Mixed Norm (LMMN)

- Constant Modulus criterion

$$J_{CM} = E[(\gamma - |\mathbf{x}^T(n)\mathbf{w}(n)|^2)^2] \tag{5}$$

Adaptive algorithm: Constant Modulus (CM)

Affine Projection Algorithm as a Projection onto an Affine Subspace

- Conditions for the analysis
  - $\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}^T(n)\mathbf{w}(n)$
  - $\mathbf{d}(n) = \mathbf{X}^T(n)\mathbf{w}^0$ defines the optimal solution in the least squares sense
  - $\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}^0$ (weight error vector)
- Error vector
  - $\mathbf{e}(n) = \mathbf{d}(n) = \mathbf{X}^T(n)[\mathbf{v}(n) + \mathbf{w}(n+1)]$
  - $= \mathbf{X}^T(n)\mathbf{w}(n+1) - \mathbf{X}^T(n)[\mathbf{v}(n) + \mathbf{w}(n+1)]$
  - $= -\mathbf{X}^T(n)\mathbf{v}(n)$

$$\mathbf{e}(n) \;=\; -\mathbf{X}^T\mathbf{v}(n) \tag{6}$$

- *Interpretation:* To minimize the error $\mathbf{v}(n)$ should be orthogonal to all input vectors
  Restriction: We are going to use only
  $\{\mathbf{x}(n),\mathbf{x}(n-1),\ldots,\mathbf{x}(n-P)\}$
- *Iterative solution:* We can subtract from $\mathbf{v}(n)$ onto the range of $\mathbf{X}n$ at each iteration
  Projection onto an affine subspace

$$\mathbf{v}(n+1) \;=\; \mathbf{v}(n) - \mathbf{P}_{\mathbf{X}(n)}\mathbf{v}(n) \tag{7}$$

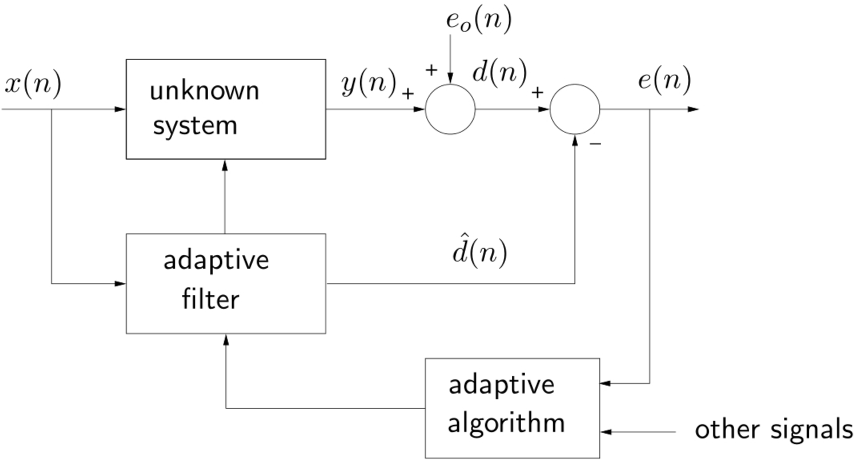
- Using $\mathbf{X}^T(n)\mathbf{v}(n) \;=\; -\mathbf{e}(n)$
  AP algorithm

$$\mathbf{v}(n+1) \;=\; \mathbf{v}(n) + \mu\mathbf{X}(n)[\mathbf{X}^T(n)\mathbf{X}(n)]^{-1}\mathbf{e}(n) \tag{8}$$

Basic Classes of Adaptive Filtering Applications

- System Identification
- Inverse System Modeling
- Signal Prediction
- Interference Cancelation

System Identification



Applications - System Identification

Channel Estimation

- Communication systems
- Objective: model the channel to design distortion compensation
- $x(n)$: training sequences

Plant Identification

- Control systems
- Objective: model the plant to design a compensator
- $x(n)$: training sequence

AffineProjection.m

API

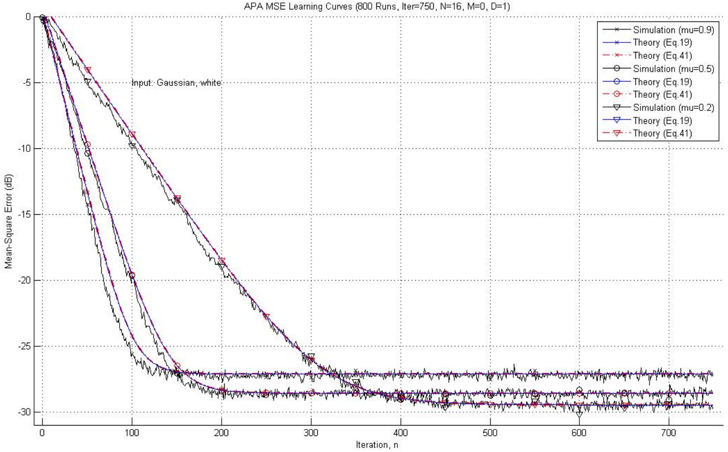
```
function    [outputVector,...
             errorVector,...
             coefficientVector] =
             AffineProjection(desired,input,S)
```

Body

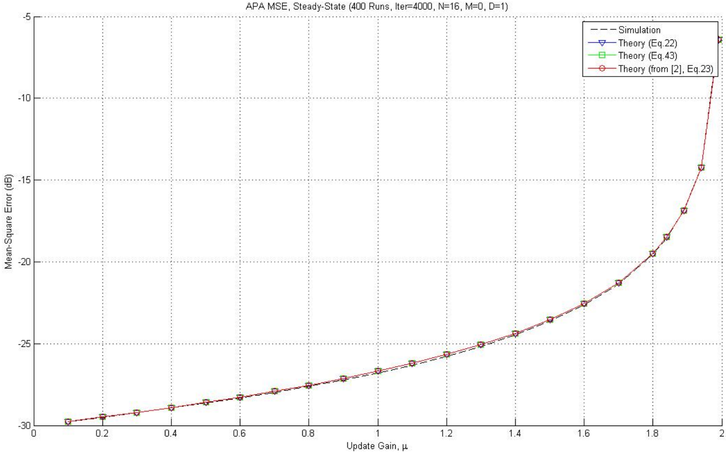
```
for it = 1:nIterations,

    regressor(:,2:S.memoryLength+1) =
    regressor(:,1:S.memoryLength);
    regressor(:,1)                  =
    prefixedInput(it+(nCoefficients-1):-1:it);
    outputVectorApConj(:,it)        =
    (regressor')*coefficientVector(:,it);
    errorVectorApConj(:,it)         =
    conj(prefixedDesired(it+(S.memoryLength):-1:it))...
    -outputVectorApConj(:,it);
    coefficientVector(:,it+1)       =
    coefficientVector(:,it)+(S.step*regressor*...
    inv(regressor'*regressor+S.gamma*...
    eye(S.memoryLength+1))*errorVectorApConj(:,it));
end
```

Learning Curve



Steady State



Weight Error Update Equations

Steepest Descent Algorithm - Stationary SOE

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}_{xx}\mathbf{w}(n)] \tag{9}$$

Newton Algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\mathbf{R}_{xx}^{-1}[-\mathbf{p} + \mathbf{R}_{xx}\mathbf{w}(n)] \tag{10}$$

Least Mean Squares (LMS) Algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \tag{11}$$

Weight Error Update Equations - continued

Normalized Least Mean Square (NLMS) Algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \frac{e(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} \tag{12}$$

Affine Projection (AP) Algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{X}(n)[\mathbf{X}^T(n)\mathbf{X}(n)]^{-1}\mathbf{e}(n) \tag{13}$$

Recursive Least Square (RLS) Algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{k}(n)\mathbf{e}(n) \tag{14}$$

Weight Error Update Equations - continued

For RLS algorithm,

$$\mathbf{k}(n) = \mathbf{P}(n)\mathbf{x}(n) \tag{15}$$

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1) \tag{16}$$

Computational Complexity

- Adaptive filter with N real coefficients and real signals
- For the AP algorithm, $K = P + 1$

Algorithm	$\times$	$+$	$/$
LMS	$2N+1$	$2N$	
NLMS	$3N+1$	$3N$	1
AP	$(K^2+2K)N+K^3+K$	$(K^2+2K)N+K^3+K$	
RLS	N^2+5N+1	N^2+3N	1

For $N = 100, P = 2$

Algorithm	$\times$	$+$	$/$	$\simeq$factor
LMS	201	200		1
NLMS	301	300	1	1.5
AP	1,530	1,536		7.5
RLS	10,501	10,300	1	52.5

Computational Complexity - continued

Typical values for acoustic echo cancellation $(N = 1024, P = 2)$

Algorithm	$\times$	$+$	$/$	$\simeq$factor
LMS	2,049	2,048		1
NLMS	3,073	3,072	1	1.5
AP	15,390	15,396		7.5
RLS	1,053,697	1,051,648	1	514

How to Deal with Computational Complexity?

★ Not an easy task!!!

★ There are "fast" versions for some algorithms (especially RLS)

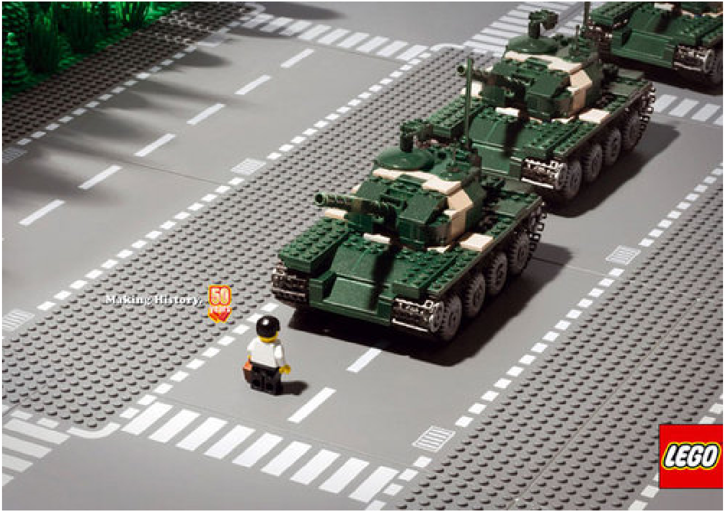
★ What is usually not said is that...speed can bring

- Instability
- Increased need for memory

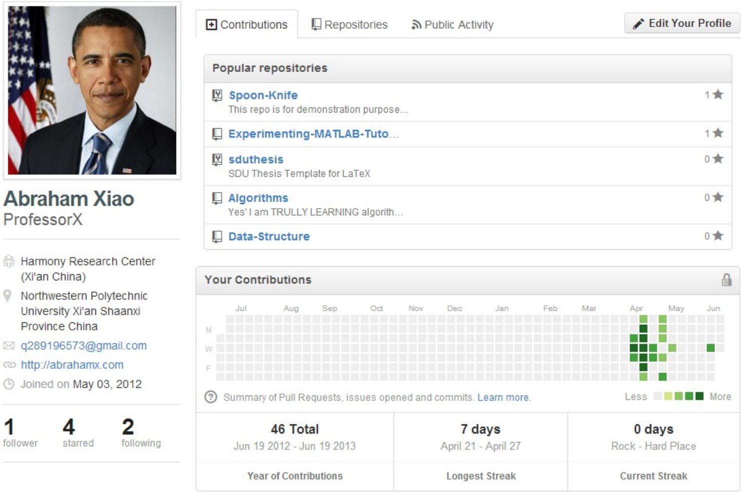
Most applications rely on simple solutions.

Instability

Memory



Workload


Contributions
Repositories
Public Activity
Edit Your Profile
Popular repositories
Spoon-Knife
This repo is for demonstration purpose...
1
Experimenting-MATLAB-Tuto...
1
sduthesis
SDU Thesis Template for LaTeX
0
Algorithms
Yes! I am TRULLY LEARNING algorith...
0
Data-Structure
0
Abraham Xiao
ProfessorX
Harmony Research Center (Xi'an China)
Northwestern Polytechnic University Xi'an Shaanxi Province China
q289196573@gmail.com
http://abrahamx.com
Joined on May 03, 2012
1 follower
4 starred
2 following
Your Contributions
Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
M
W
F
Summary of Pull Requests, issues opened and commits. Learn more.
Less More
46 Total
Jun 19 2012 - Jun 19 2013
Year of Contributions
7 days
April 21 - April 27
Longest Streak
0 days
Rock - Hard Place
Current Streak


Any questions?