**Machine Learning for Signal Processing**

**[5LSL0]**

**Assignment 1: Optimum Linear Filters**

**REPORT**

**Names including ID:**  
 **1: Jie Zhang (1616528)**  
 **2: Rundong Zhang (1554301)**

**Date: 2021/05/04**

## Known statistics

### Wiener filter



=



When equals the optimal Wiener filter , equals 0. This means that .



Use time-averaging to estimate the auto correlation and the cross correlation. If the data set is very large, then the computation will become quite expensive since it needs to sum all the elements.

### Steepest gradient descent

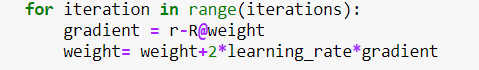


Gradient descent algorithm reaches to steady state when the difference weight vector is equal to zero after enough recursion. Thus, the value of weights are equal to the Winer filter weights.



**Thus, GD algorithm stable if =0.143**

1. GD filter update Python code (insert only the relevant lines)

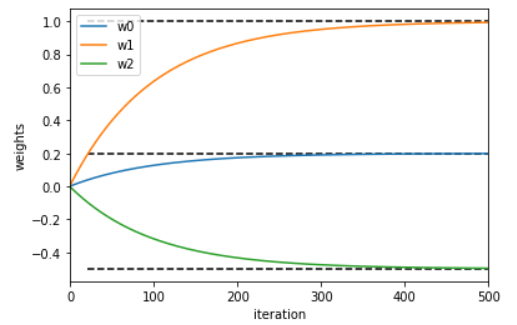


|  |
| --- |
| *Insert plot and comments here*    Notice that the iteration is actual 10000 times, here cut off the rest part for convenience. Because the convergence stops quickly. Same rules for the plot of the other method.    **Comment on the convergence of w0 and w1:**  As it is calculated by hand above, w0 and w1 convergence to the value of Wiener filter weights. The learning rate is set to 0.005 which is satisfied with the requirement for stable state as well.  Since it sums up over all examples on each iteration when performing the updates to the parameters, the trajectory is straightly (no noise) towards to the minimum and guaranteed to the converge if it is stable. |

### Newton algorithm



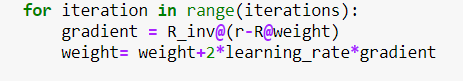
Once the learning rate is fixed, the change of variables will be constant. It is also shown in the figure below that all the weights converge at the same rate.





So, the Newton’s method is stable when

1. Newton filter update Python code (insert only the relevant line)



|  |
| --- |
| *Insert plot and comments here* |

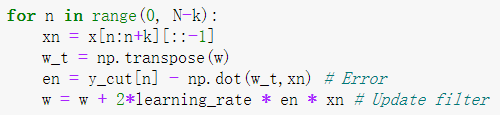
***Comments***

Newton’s method also achieves the convergence with the correct values. As we can see the Newton’smethod has a straight line in the contour plot, comparing to the GD algorithm. That’s because Newton’s method approximates the curve with its local quadratic which means it can make more efficient updates per step. Therefore, we can see in the contour, it takes a more direct route.

## Unknown statistics

### (N)LMS





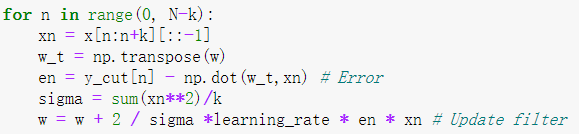
|  |
| --- |
| *Insert plot here* |

***Comments***

LMS method achieves the convergence around the correct values. As we can see in the above figures, the weights converge quickly to the optimal with oscillation. And it keeps oscillating around the optimal weights.

Trade-off choosing : If is too large, it can hardly converge to the optimal weights, and oscillate around it. While, if it is too small, the calculation time would be too long. So, its better to choose a smaller to first guarantee the result accuracy. Here we choose 0.005, and with this step size the calculation time is acceptable.





|  |
| --- |
| *Insert plot here* |
| ***Comments***  NLMS method also achieves the convergence around the correct values, but slower compared with LMS methods with the same . While, the oscillation of NLMS is smaller. So, the result accuracy is higher than LMS method, as shown in the above figures. |

### RLS

RLS finds efficient recursive solution for LS problem in each iteration. It iterates R and r matrix by minimizing the error square mean. In this method, we sequent obtain the data, so the weight vector w is kept updating. And it introduces a forgetting factor to evaluate the impact of data on the current model, so that the later the data, the greater the impact. Thus, the results converge fast and the accuracy is better than other algorithms.

The forgetting factor influences on the convergence accuracy and stability. The bigger factor, the better accuracy and stability. As shown below, in the first figure the forgetting factor is 1-10-4. The second one is 1-10-2.

|  |
| --- |
| *Insert plot here*   1. *Forgetting factor = 0.9999*      1. *Forgetting factor = 0.99* |

|  |  |  |
| --- | --- | --- |
|  | Computational complexity | Convergence speed/stability/accuracy |
| LMS | #1 | #3 |
| NLMS | #2 | #2 |
| RLS | #3 | #1 |

*Comments*

LMS is a simple method that has less computational complexity, but its convergence speed is not good and the accuracy is actual worst. Normalized LMS applies an improvement to LMS method, so its complexity is a little higher, while the accuracy and stability is better. The RLS algorithm belongs to the least squares algorithm, so has best stability, speed and accuracy. When it iterates infinite times, it could converge to Wiener Filter. However, its complexity is biggest because it has more matrix calculation.