Parameter Optimization for Step-Adaptive Approximate Least Squares

M. Lunglmayr(✉) and M. Huemer

Institute of Signal Processing, Johannes Kepler University Linz, Linz, Austria
Michael.Lunglmayr@jku.at

Abstract. We discuss possible approaches for the adaption of the step width of Step-Adaptive Approximate Least Squares. We present and compare two low complexity and practically feasible adaptation functions whose parameters have been optimized based on computer simulations. We show that by applying these approaches the performance deviation of Step-Adaptive Approximate Least Squares lies within the single percentage range compared to the optimal least squares solution.

Keywords: Estimation · Least squares · Approximate least squares · Implementation

1 Introduction

The linear least squares (LS) estimation approach is an important concept in electrical engineering, especially in digital signal processing. Example applications range from localization [1] and positioning [2], over robotics [3], power and battery applications [4], biomedical applications [5], as well as image processing [6].

For LS estimation we assume the following system model:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{y}$ is a measured vector, $\mathbf{H}$ is a known system matrix of dimension $m \times p$, $\mathbf{n}$ is a noise vector and $\mathbf{x}$ is the parameter vector that we want to estimate.

The linear least squares solution to this estimation problem is well known as

$$\hat{\mathbf{x}}_{LS} = \mathbf{H}^{\dagger}\mathbf{y}. \tag{2}$$

with the pseudoinverse $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$. Such a direct calculation of the solution as is often called *batch solution* in literature [7].

For many realtime applications a low complexity implementation is preferred over an exact solution. For this reason the calculation of the batch solution is usually avoided for such applications, due to its computational complexity and its large memory requirements. To provide a low complexity approximate approach for the LS estimation problem we developed a method that we call *Approximate*

R. Moreno-Díaz et al. (Eds.): EUROCAST 2015, LNCS 9520, pp. 521–528, 2015.
DOI: 10.1007/978-3-319-27340-2_65

Least Squares (ALS) [8]. It can be seen as a variant of the Kaczmarz algorithm [9] for overdetermined and inconsistent linear equation systems.

ALS is based on the iterative least squares (ILS) approach that iteratively calculates

$$\hat{\mathbf{x}}^{(k)} = \hat{\mathbf{x}}^{(k-1)} - \mu \mathbf{d}(\hat{\mathbf{x}}^{(k-1)}). \tag{3}$$

Here $\mathbf{h}_i^T$ is the i^{th} row of $\mathbf{H}$. The function

$$\mathbf{d}(\mathbf{x}^{(k-1)}) = \sum_{i=1}^{m} 2\mathbf{h}_i(\mathbf{h}_i^T \hat{\mathbf{x}}^{(k-1)} - y_i) \tag{4}$$

is the gradient of the least squares cost function

$$J(\hat{\mathbf{x}}) = \sum_{i=1}^{m} (y_i - \mathbf{h}_i^T \hat{\mathbf{x}})^2 \tag{5}$$

that has its minimum at $\hat{\mathbf{x}}_{LS}$. It can be shown that for $k \to \infty$, $\hat{\mathbf{x}}^{(k)}$ converges to $\hat{\mathbf{x}}_{LS}$ given that the iteration step width μ fulfills $0 < \mu < 1/(2s_1^2(\mathbf{H}))$ [12], with $s_1(\mathbf{H})$ as the largest singular value of $\mathbf{H}$. Figure 1 schematically shows one iteration of ILS. The gradient $\mathbf{d}(\mathbf{x}^{(k-1)})$ can be seen as a sum of partial gradients

$$d_i(\hat{\mathbf{x}}^{(k-1)}) = 2\mathbf{h}_i(\mathbf{h}_i^T \hat{\mathbf{x}}^{(k-1)} - y_i). \tag{6}$$

2 Approximate Least Squares

To reduce the complexity of this approach we proposed to use only one of these partial gradients per iteration, leading to the ALS iteration:

$$\hat{\mathbf{x}}^{(k)} = \hat{\mathbf{x}}^{(k-1)} + 2\mu \mathbf{h}_{k\urcorner}(y_{k\urcorner} - \mathbf{h}_{k\urcorner}^T \hat{\mathbf{x}}^{(k-1)}). \tag{7}$$

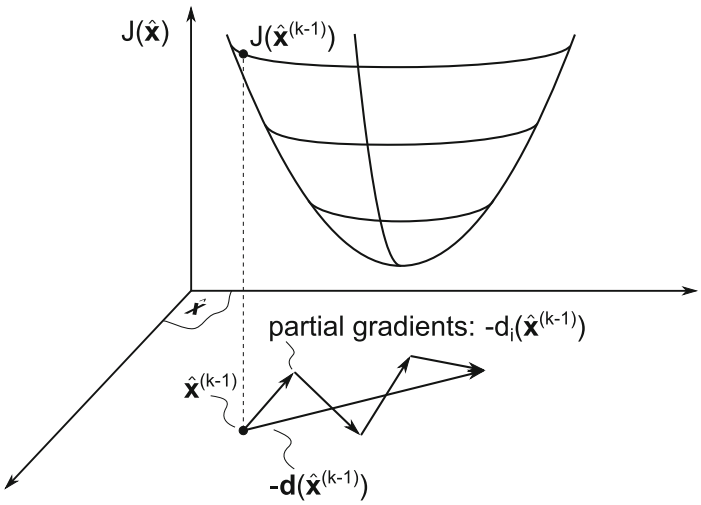

Fig. 1. Schematical drawing of the gradient and partial gradients.

In the above equation the operator " $\urcorner$" is defined as: $k^{\urcorner} = ((k-1) \bmod m) + 1$ for a positive natural number k. For better readability we don't write the dependence of this operator on m in the operator's symbol. For ALS m is always the number of rows of the matrix $\mathbf{H}$. Using this operator naturally allows to have more iterations than the number of rows of $\mathbf{H}$, which is typically required for ALS to obtain a good performance.

As we described in [8], the error $\mathbf{e}^{(k)} = \hat{\mathbf{x}}^{(k)} - \mathbf{x}$ of the above iteration can be split into two parts

$$\mathbf{e}^{(k)} = \mathbf{e}_0^{(k)} + \mathbf{e}_{\boldsymbol{\Delta}}^{(k)}, \tag{8}$$

with $\mathbf{e}_0^{(k)}$ as the error depending on the initial value $\hat{\mathbf{x}}^{(0)}$ of the algorithm before the first iteration, and $\mathbf{e}_{\boldsymbol{\Delta}}^{(k)}$ as the error depending on the noise vector $\mathbf{n}$. As we showed in [8], the error $\mathbf{e}_0^{(k)}$ goes to zero as the number of iterations goes to infinity, while the noise dependent error $\mathbf{e}_{\boldsymbol{\Delta}}^{(k)}$ persists even if $k \to \infty$. Due to the cyclic re-use of the rows of $\mathbf{H}$ as well as the measurement values in $\mathbf{y}$, the errors $\mathbf{e}^{(k)}$ converge to m different values for $k \to \infty$:

$$\mathbf{e}^{(mk+i)} = \mathbf{e}_{\boldsymbol{\Delta}}^{(mk+i)} = \mathbf{e}_{\boldsymbol{\Delta}_\infty}^{(i)} \text{ for } i = 1, \ldots, m \tag{9}$$

In Fig. 2 we plotted a typical case of the error norm of $\hat{\mathbf{x}}^{(k)}$ for an example 100×10 matrix $\mathbf{H}$ over the iterations k as well as the error norm of the algorithm's output $\hat{\mathbf{x}}_{ALS}$, as described below. For better visibility, the error norm of $\hat{\mathbf{x}}_{ALS}$ has been depicted has a horizontal line, although it is available only at the end of the algorithm. When analyzing the norm of the error $\mathbf{e}^{(k)}$ over the iterations, one can see an oscillatory behavior. This comes from the fact that for large k the ALS algorithm produces approximately the same m recurring error vectors (up to the vanishing deviation $\mathbf{e}_0^{(k)}$). To reduce the final error norm of the vector output by the ALS algorithm, we introduced an averaging step in the final m iterations of the algorithm.

The basic ALS algorithm is summarized in the following pseudocode (Algorithm: ALS).

Algorithm: ALS

$\hat{\mathbf{x}}_{ALS} = \mathbf{0}$
$\hat{\mathbf{x}}^{(0)} = \mathbf{0}$
for $k = 1, \ldots, N$ **do**
  $\hat{\mathbf{x}}^{(k)} = \hat{\mathbf{x}}^{(k-1)} + \mu 2 \mathbf{h}_{k^{\urcorner}} (y_{k^{\urcorner}} - \mathbf{h}_{k^{\urcorner}}^T \hat{\mathbf{x}}^{(k-1)})$
  if $k > N - m$ **then**
    $\hat{\mathbf{x}}_{ALS} = \hat{\mathbf{x}}_{ALS} + \hat{\mathbf{x}}^{(k)}$
  end if
end for
$\hat{\mathbf{x}}_{ALS} = \frac{1}{m} \hat{\mathbf{x}}_{ALS}$

Here N denotes the number of iterations of the algorithm and $\hat{\mathbf{x}}_{ALS}$ is the approximation of $\hat{\mathbf{x}}_{LS}$ that is output by the algorithm. $\mathbf{0}$ denotes the zero vector.

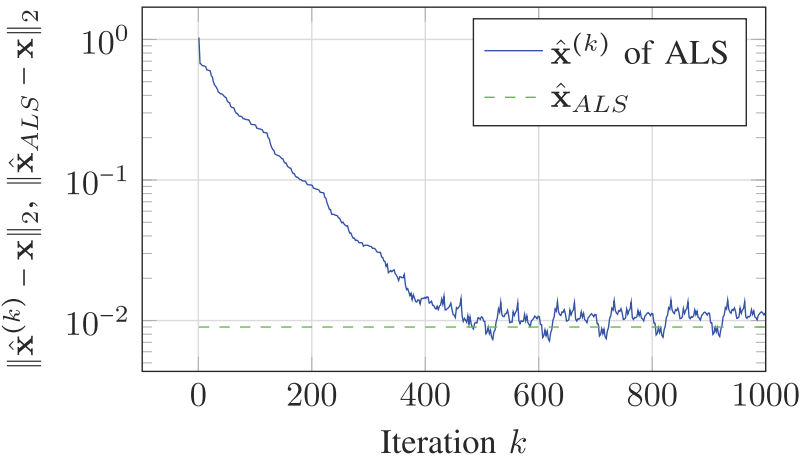

Fig. 2. ALS error norms.

As one can see from the algorithm's description the averaging only has to be done *once*, it therefore presents only a minor complexity increase (overall only pm additions and p multiplications with the constant $1/m$ have to be performed additionally).

As one can see in the above algorithm, if k reaches m, then for the following iterations the first rows of $\mathbf{H}$ and the first elements of $\mathbf{y}$ are re-used again in a cyclic manner. This approach has the advantage, that compared to ILS, about m times less multiplications per iteration are required.

But ALS also has the drawback – as we show in [8] and as discussed above – that for a constant step width μ, a persistent oscillating error $\mathbf{e}^{(k)} = \hat{\mathbf{x}}^{(k)} - \mathbf{x}_{LS}$ exists, even if $k \to \infty$. This error depends on the noise as well as on the parameter μ of the algorithm. While a large value of μ leads to a fast decrease of the error $\mathbf{e}_0^{(k)}$ at early iterations it leads to a higher error $\mathbf{e}_{\mathbf{\Delta}}^{(k)}$ and thus to a high final error. Choosing small values leads to small final errors but requires a large number of iterations because the error $\mathbf{e}_0^{(k)}$ is decreasing more slowly. Supporting these findings, in [10] it has been shown that the ALS iteration converges to the LS solution, if $\mu \to 0$ and $k \to \infty$. However, for a practical application of the algorithm, one is interested in a reduction method providing a fast convergence close to $\mathbf{x}_{LS}$.

For this reason we proposed to adjust the step width μ of the algorithm during the iterations. We call this approach Step-Adaptive Approximate Least Squares (SALS) [11].

3 Step-Adaptive Approximate Least Squares (SALS)

In [11] we proposed to adjust the step width $\mu = \mu_k$ at every iteration. For this we divide the overall ALS iteration process into two phases, the *reduction phase*

and the *oscillation phase*. In the reduction phase the error norm $\|\mathbf{e}^{(k)}\|_2$ decreases while in the oscillation phase the error norm is approximately cyclically repeating as described above. In Fig. 2 the reduction phase lasts the first 500 iterations. During the reduction phase, the error part $\mathbf{e}_0^{(k)}$ (due to the initial value $\mathbf{x}^{(0)}$) contributes most to $\mathbf{e}^{(k)}$ in (8), thus a high value of μ_k should be used to decrease the overall error. In the oscillation phase, the error $\mathbf{e}_{\boldsymbol{\Delta}}^{(k)}$ contributes most to $\mathbf{e}^{(k)}$. Here a high value of μ would prevent a further decrease of the error, thus a low value of μ_k is beneficial in this phase. For the reduction phase we propose, to use

$$\mu_k = \frac{1}{2\|\mathbf{h}_{k^\neg}^T\|_2^2}. \tag{10}$$

As one can show [11], this is the largest value of μ for which $\mathbf{e}_0^{(k)} \to \mathbf{0}$ as $k \to \infty$. To detect whether or not the algorithm is already in the oscillation phase we used the following method. The oscillation phase is characterized by the occurrence of approximately the same error vectors, every m iterations. When inspecting (7) one can see that the error vector is only influenced by the term $(y_{k^\neg} - \mathbf{h}_{k^\neg}^T \hat{\mathbf{x}}^{(k-1)})$. By comparing this value with the value m iterations before one can detect the oscillation phase if the difference is below a predefined threshold. After the oscillation phase is detected, the idea for SALS is to reduce

Algorithm: SALS

$\hat{\mathbf{x}}_{SALS} \leftarrow \mathbf{0}$
$\hat{\mathbf{x}}^{(0)} \leftarrow \mathbf{0}$
$v_k \leftarrow 0$
$v_{k-m} \leftarrow 1$
DontReduceMu $\leftarrow$ True
for $k = 1 \ldots N$ **do**
  $v_k \leftarrow y_{k^\neg} - \mathbf{h}_{k^\neg}^T \hat{\mathbf{x}}^{(k-1)}$
  if DontReduceMu **then**
    $\mu \leftarrow \mu_{k^\neg}$ according to (10)
    if $k^\neg = 1$ **then**
      if $|v_k - v_{k-m}| < v_{th}$ **then**
        DontReduceMu $\leftarrow$ False
        $\mu \leftarrow \frac{1}{2 \max\limits_{i=1\ldots m} \|\mathbf{h}_i^T\|_2^2}$
      end if
      $v_{k-m} \leftarrow v_k$
    end if
  else
    $\mu \leftarrow f(\mu)$
  end if
  $\hat{\mathbf{x}}^{(k)} \leftarrow \hat{\mathbf{x}}^{(k-1)} + \mu 2 \mathbf{h}_{k^\neg} v_k$
  if $k > N - m$ **then**
    $\hat{\mathbf{x}}_{SALS} \leftarrow \hat{\mathbf{x}}_{SALS} + \hat{\mathbf{x}}^{(k)}$
  end if
end for
$\hat{\mathbf{x}}_{SALS} \leftarrow \frac{1}{m} \hat{\mathbf{x}}_{SALS}$

the step width μ to as well reduce the final error of the ALS solution. The overall SALS algorithm is presented in the following pseudocode (Algorithm: SALS). The function $f(\mu)$ is used to reduce μ once the oscillation phase is detected. The first reduction is done by setting μ to the minimum of the m values used in the iterations before. In every following iteration μ is furthermore reduced via a reduction function $f(\mu)$. In the following section we present and compare two low complexity variants of $f(\mu)$.

4 Simulation Results

The aim of Approximate Least Squares is to provide a low complexity approximate solution of the linear least squares problem. For this reason we restrict ourselves to reduction functions causing only a negligible complexity overhead compared to the basic ALS algorithm. We specifically compare the two reduction functions

$$\mu_k = \mu_{k-1} - s = f_1(\mu_{k-1}) \tag{11}$$

as well as

$$\mu_k = (1 - 2^{-c})\mu_k = f_2(\mu_{k-1}), \tag{12}$$

with a positive fractional s and a positive integer c. For (11) only one subtraction per iteration is required, while for (12) only one shift operation and a subtraction is required per iteration. Figure 3 shows simulation results for random $\mathbf{H}$ matrices for ALS and SALS using these reduction functions, respectively. The entries of these matrices have been sampled from a uniform distribution out of $[0, 1]$. Every simulation has been done for white Gaussian noise with zero mean and standard deviation $\sigma \in S = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, respectively. In the SALS algorithms v_{th} was set to 10^{-6}.

For (11), we used $s = \mu_i/(N - i)$, with i as the iteration when the oscillation phase was detected and μ_i as the corresponding step width, respectively. For (12), c was set to $\lfloor log_2(N) \rfloor$. These values of s and c have been found and optimized by extensive simulations, respectively.

The figure shows the relative error of ALS and SALS, respectively, compared to the optimum least squares solution. It shows the maximum relative increase of the error norms of ALS and SALS, respectively, over the averaged error norms of LS. The maximization has been done over the elements of S: $r_{ALS} = \max_S \left(\frac{||\hat{\mathbf{x}}_{ALS} - \mathbf{x}||_2}{||\hat{\mathbf{x}}_{LS} - \mathbf{x}||_2} - 1 \right)$ and $r_{SALS} = \max_S \left(\frac{||\hat{\mathbf{x}}_{SALS} - \mathbf{x}||_2}{||\hat{\mathbf{x}}_{LS} - \mathbf{x}||_2} - 1 \right)$, respectively. As one can see from these results, SALS performs significantly better than the basic ALS algorithm. Its deviation to the least squares error norm is within the single percentage range. For large $\mathbf{H}$ matrices f_1 performs significantly better than f_2. For the simulated matrices with 1000 rows, the average error norm deviation r_{SALS} was below 2 %. For a practical application, especially when thinking of an implementation in fixed point precision, such an error deviation is typically considered negligible.

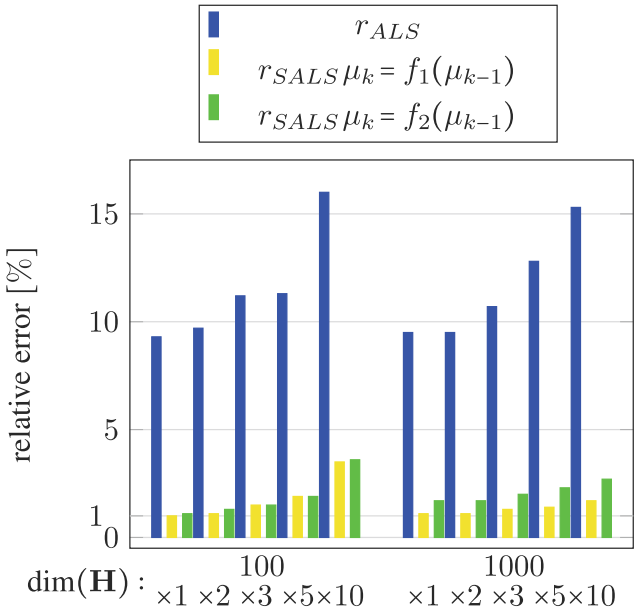

Fig. 3. Simulation results.

5 Conclusion

We present and compare different low complexity approaches for reducing the step width of Step-Adaptive Approximate Least Squares. We show that with Step-Adaptive Approximate Least Squares an error norm performance can be achieved that is within the single percentage range compared to the optimal least squares error performance. For many applications such an error norm can be considered practically equivalent to the optimum least squares solution that is obtainable only with a much higher computational effort.

References

1. Choi, K.H., Ra, W.-S., Park, S.-Y., Park, J.B.: Robust least squares approach to passive target localization using ultrasonic receiver array. IEEE Trans. Ind. Electron. **61**(4), 1993–2002 (2014)
2. Thomas, R.R., Maharaj, B.T., Zayen, B., Knopp, R.: Multiband time-of-arrival positioning technique using an ultra-high-frequency bandwidth availability model for cognitive radio. IET Radar Sonar Navig. **7**(5), 544–552 (2013)
3. Gautier, M., Janot, A., Vandanjon, P.-O.: A new closed-loop output error method for parameter identification of robot dynamics. IEEE Trans. Control Syst. Technol. **21**(2), 428–444 (2013)
4. Unterrieder, C., Zhang, C., Lunglmayr, M., Priewasser, R., Marsili, S., Huemer, M.: Battery state-of-charge estimation using approximate least squares. J. Power Sources **278**, 274–286 (2015)

5. Yuqian, L., Sima, D.M., Van Cauter, S., Himmelreich, U., Sava, A.C., Yiming, P., Yipeng, L., Van Huffel, S.: Unsupervised nosologic imaging for glioma diagnosis. IEEE Trans. Biomed. Eng. **60**(6), 1760–1763 (2013)
6. Rouhani, M., Domingo Sappa, A.: The richer representation the better registration. IEEE Trans. Image Process. **22**(12), 5036–5049 (2013)
7. Kay, S.M.: Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice Hall, Englewood Cliffs (2005)
8. Lunglmayr, M., Unterrieder, C., Huemer, M.: Approximate least squares. In: 2014 Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), pp. 4678–4682 (2014)
9. Kaczmarz, S.: Przyblizone rozwiazywanie ukladw rwnan liniowych. Angenäherte Auflösung von Systemen linearer Gleichungen. In: Bulletin International de lAcadmie Polonaise des Sciences et des Lettres. Classe des Sciences Mathmatiques et Naturelles, Srie A, Sciences Mathmatiques, pp. 355–357 (1937)
10. Censor, Y., Eggermont, P.P.B., Gordon, D.: Strong underrelaxation in Kaczmarz's method for inconsistent systems. Numer. Math. **41**(1), 83–92 (1983)
11. Lunglmayr, M., Unterrieder, C., Huemer, M.: Step-adaptive least squares. Submitted to European Signal Processing Conference (EUSIPCO) (2015)
12. Björck, A.: Numerical Methods for Least Squares Problems. SIAM Philadelphia, Philadelphia (1996)