An Alternating Least Squares Algorithm with Application to Image Processing

Lorenzo Piazzo
University of Rome
DIET dept.
V. Eudossiana 18
Italy
lorenzo.piazzo@uniroma1.it

Abstract: Least Squares (LS) estimation is a classical problem, often arising in practice. When the dimension of the problem is large, the solution may be difficult to obtain, due to complexity reasons. A general way to reduce the complexity is that of breaking the problem in smaller sub-problems. Following this approach, in the paper we introduce an Alternating Least Squares (ALS) algorithm that finds the LS estimate by iteratively solving two sub-problems. The algorithm can speed up the solution of any LS problem, but it is especially well suited for applications where the partition arises naturally so that the sub-problems have a structure and are simple to solve. To illustrate this fact, we discuss the application of the ALS to an image formation system affected by noise and drift, describing an efficient implementation and showing that the ALS is an effective image formation method.

Key–Words: Least Squares, Image Formation

1 Introduction

The Least Squares (LS) estimation problem, e.g. [1], often arises in practice and is important for many applications. Indeed, this is a classical problem and a number of established solution methods exist. However, when the dimension of the problem is large, as occurs in practical applications, the solution may be difficult to obtain, due to complexity reasons, and the development of efficient algorithms and approaches is an open and challenging issue.

A general way for reducing the solution complexity is that of breaking the LS problem into smaller sub-problems, e.g. [2]. Following this approach, in the paper we present and discuss an algorithm that computes the solution of the full problem by iteratively solving two sub-problems. The algorithm falls in the class of the Alternating Least Squares (ALS) methods [3] and is similar but not identical to the method proposed in [4].

While the ALS algorithm can be used to simplify the solution of any large LS problem, there are specific applications where the partition into sub-problems arises naturally and the sub-problems have a structure that makes their solution simple. This is the case when the ALS approach is most useful and can significantly speed up the solution. To illustrate this fact, in the second part of the paper we present an example of application. Specifically, we consider an image acquisition system where the detectors are affected by noise and by a second disturbance, called a drift, which is a slow deviation of the readouts from the baseline level. Such a disturbance is found in several practical systems, including astronomical imaging [5], medical imaging [6] and electron microscopy [7]. In this context, we discuss the image synthesis and show that the ALS is an efficient way of computing the LS estimate. Note that the same application is also discussed in [4]. However, in [4] the drift is assumed to be a polynomial curve while in this paper we assume that it is a generic low-pass signal, which is a more general and practical model.

The paper is organised as follows. In section 2 we present the LS problem and review some basic concepts. In section 3 we introduce and discuss the ALS algorithm. In section 4 we describe an image acquisition system affected by drift and specialise the ALS to this case, showing that it is an effective way to carry out the LS image synthesis. In section 5 we present some experimental results and in section 6 we give the conclusions.

Notation. In the paper we use lowercase letters to denote vectors and uppercase letters to denote matrices, e.g. $v, A$. We use a subscript to denote an element of a vector or a matrix, e.g. $v_k, A_{i,j}$. We use a superscript $T$ to denote matrix or vector transposition, e.g. $A^T$. 
2 Preliminaries

Consider an $H \times 1$ vector $z$, representing the parameters of some physical system. Suppose that the system is observed by means of a linear, noisy instrument, which produces $D >> H$ readouts, represented by a $D \times 1$ data vector

$$d = Az + n$$

where $n$ is a $D \times 1$ noise vector, the elements of which are a set of independent, identically distributed (iid), zero mean random variables, and $A$ is a $D \times H$ known, full rank, matrix, which depends on the instrument and on the observation strategy. Given this set up, an important problem is that of producing an estimate of $z$ from the knowledge of $d$ and $A$. This is a classical problem having several established solutions. One of the most important is the Least Squares (LS) estimate, which is obtained by solving in the LS sense the over-determined linear system $Az = d$ associated with model (1). The LS estimate is given by [1]

$$\hat{z} = (A^T A)^{-1} A^T d$$

where $(A^T A)^{-1} A^T$ is known as the pseudo-inverse of $A$. Strictly related to the LS estimate is the vector

$$\bar{d} = A \hat{z}$$

which will be called the projection of the solution and represents the output of a noiseless instrument when $z = \hat{z}$.

Equation (2) gives the LS estimate explicitly but is difficult to compute directly when the problem dimension is large. The most challenging step is the inversion of the $H \times H$ matrix $A^T A$, which becomes impossible when $H$ is higher than, say, a few thousands. Larger problems have to be solved using iterative methods, like the Parallel Conjugate Gradient (PCG) [8], which yield an approximate yet normally accurate solution.

A general way to reduce the computational complexity is that of breaking the problem into smaller sub-problems. In particular, we can partition the elements of the parameter vector into two sub-vectors, writing $z = [m^T, a^T]^T$ where $m$ is an $M \times 1$ vector and $a$ is a $K \times 1$ vector, with $M + K = H$. Next, we make a similar partition on the columns of the matrix $A$ and write $A = [P, X]$ where $P$ is a $D \times M$ matrix and $X$ is a $D \times K$ matrix. In this way the data model of equation (1) can equivalently be written as

$$d = P m + X a + n.$$  \hfill (3)

Moreover, using a block matrix pseudo-inverse formula [2], the LS estimate of equation (2) can be written as

$$\bar{z} = \begin{bmatrix} \bar{m} \\ \bar{a} \end{bmatrix} = \begin{bmatrix} (P^T \Pi_X^\perp P)^{-1} P^T \Pi_X^\perp \\ (X^T \Pi_X^\perp X)^{-1} X^T \Pi_X^\perp \end{bmatrix} d$$

where $\Pi_X^\perp$ is the projector onto the orthogonal complement of the subspace spanned by the columns of $P$ and $\Pi_X^\perp$ is the projector onto the orthogonal complement of the subspace spanned by the columns of $X$.

Equation (4) is useful because it allows to separately compute the two parts of the estimate. For example, the vector $\bar{m}$ is

$$\bar{m} = (P^T \Pi_X^\perp P)^{-1} P^T \Pi_X^\perp d.$$  \hfill (4)

Moreover, the computation of $\bar{m}$ requires the inversion of the $M \times M$ matrix $P^T \Pi_X^\perp P$. Similarly, the computation of $\bar{a}$ requires the inversion of the $K \times K$ matrix $X^T \Pi_X^\perp X$. Therefore, by partitioning, we replaced the inversion of an $H \times H$ matrix with the inversion of smaller matrices, which may yield a significant simplification. In the next section we present another approach that achieves similar advantages.

3 Alternating Least Squares Algorithm

Based on the model of equation (3), we can consider two additional over-determined linear systems, namely $P m = d$ and $X a = d$. By solving the first system in the LS sense we obtain a vector

$$\bar{m} = (P^T P)^{-1} P^T d$$

which is the LS estimate of $m$ when $a = 0$. Clearly, since in general $a \neq 0$, this is not a good estimate and, in particular, it is not the LS estimate. However, the latter expression, together with a corresponding one for the second system, can be used as the basis of an algorithm that iteratively finds the LS estimate and is described in the following.

The algorithm is an Alternating Least Squares (ALS) scheme, constituted by the following steps

0. Set $\delta = d$. Repeat 1-4 until convergence:
1. $\hat{m} = (P^T P)^{-1} P^T \delta$.
2. $\eta = d - P \hat{m}$.
3. $\hat{a} = (X^T X)^{-1} X^T \eta$.
4. $\delta = d - X \hat{a}$.

The algorithm maintains two vectors, namely $\hat{m}$ and $\hat{a}$, which are the current estimates. At each iteration,
the vectors are updated, in steps 1 and 3, by solving two LS sub-problems, namely those associated with the linear systems \( P\hat{m} = \delta \) and \( X\hat{a} = \eta \). The key point is that the known vectors of the sub-problems, namely \( \delta \) and \( \eta \), are obtained from the known vector of the full problem, \( d \), by appropriately subtracting the projection of the current estimates.

It can be shown that the ALS algorithm converges, since the vectors \( \hat{m} \) and \( \hat{a} \) tend towards a limiting value. Moreover, at convergence, we have \( \hat{m} = \hat{m} \) and \( \hat{a} = \hat{a} \). Therefore, the ALS converges towards the LS estimate. The proof is based on the Alternating Projection Theorem [9] and can be found\(^2\) in [4].

The ALS algorithm requires the inversion of an \( M \times M \) matrix, namely \( P^T P \), and of a \( K \times K \) matrix, namely \( X^TX \). Therefore, it avoids the inversion of a full \( H \times H \) matrix. Moreover, there are practical cases when the partition into sub-problems arises naturally and the resulting \( P \) and \( X \) matrices have a simple structure. In this case the inversion can be performed analytically or inexpensively and the ALS algorithm is an excellent choice, which can significantly speed up the computation of the LS estimate. We give an example in the next section.

To conclude, we note that the algorithm can easily be extended to the case of a partition into three or more sub-problems and to the case when the noise is not white. Moreover, note that in this paper we only consider the case when the columns of the \( P \) and \( X \) matrix are linearly independent, as is implied by the assumption that \( A \) is full rank. However, the ALS algorithm can be used in the more general case when there is linear dependency between the matrices: the analysis is more involved and can be found in [4].

4 Application to image processing

As an application of the ALS, we discuss how the algorithm can be used to filter out the low frequency drift which affects some image acquisition systems. To be specific, we consider a simplified model of the PACS instrument [10], which is a photometer onboard the European Space Agency (ESA) Herschel space telescope [11]. However, the model applies equally well to other imaging systems.

The PACS photometer consists of an array of 2048 detectors (bolometers), tuned on a narrow band of the infrared spectrum. The field of view of the array is approximately 45 squared arcsec, but a typical observation covers a much larger sky area, up to some square degrees wide. To observe the area, the telescope pointing is moved along a set of parallel scan lines, covering the area. During the observation, the detectors are sampled at frequency \( f_s = 40 \) Hz, producing a sequence of readouts for each detector which is termed a timeline. The timelines, together with the corresponding pointing information, constitute the observation raw output.

The timelines are affected by several disturbances, see [5] for a list, but in our simplified example we only consider two. In particular, we assume that the readouts are affected by Additive White Gaussian Noise (AWGN). Moreover, we assume that the timelines are affected by a drift, i.e. a slow deviation from the baseline, which is modelled as an unknown, deterministic, low-pass signal. In particular, we assume that the drift spectrum is contained in the band \([-f_d/2, f_d/2]\) where \( f_d << f_s \) is the drift bandwidth.

4.1 Data model

We now develop a mathematical model for the PACS data. As a starting point, we stack all the timelines into a vector \( d \). Specifically, if the output is composed of \( N_t \) timelines each composed of \( N_r \) readouts, \( d \) is a \( D \times 1 \) vector where \( D = N_t N_r \) is the total number of readouts. Next, we write the vector \( d \) as the sum of three components, namely

\[
d = s + y + n
\]

where \( s \) is the signal vector, representing the useful part of the readout, measuring the emission, \( y \) is the drift vector, representing the low frequency deviation from the baseline, and \( n \) is the noise vector, representing the AWGN. The noise vector is a sequence of iid, zero mean, Gaussian random variables. The signal and drift vectors are discussed in the following.

In order to develop an expression for the signal vector, we assume that the sky is pixellised, i.e. that it is partitioned into a grid of \( M \) non-overlapping squares (pixels) where the flux is constant, and is represented by an \( M \times 1 \) vector \( m \). Next, we assume that the signal component of each readout is equal to the value of the sky pixel where the detector was pointed at the sampling time\(^3\). Then, the signal vector can be written as

\[
s = Pm
\]

where \( P \) is a \( D \times M \), sparse, binary matrix constructed from the pointing information and such that \( P_{k,i} = 1 \) if the \( k \)-th readout falls into the \( i \)-th sky pixel and

\(^2\)In [4] a different version of the ALS algorithm is presented and analysed. However the two versions are equivalent and the proof can be adapted.

\(^3\)The model is clearly an approximation, since it neglects the telescope Point Spread Function (PSF) and the continuous nature of the sky. However, the approximation is normally good for PACS data. See [4] for a deeper discussion.
where \( f(t) = \sum_{k=1}^{N_a} a_k \text{sinc}\left(\frac{t-kT}{T}\right) \) \(^{(8)}\)

where \( \text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \). When the detector is sampled, the drift is sampled too, to produce a sequence of \( N_r \) samples, which we denote by \( y_i = f(iT_s) \) for \( i = 1, ..., N_r \), where \( T_s = 1/f_s \) is the sampling spacing. Using equation (8), these samples can be written as

\[
y_i = \sum_{k=1}^{N_a} a_k \text{sinc}\left(\frac{iT_s-kT}{T}\right) \quad \text{for } i = 1, ..., N_r.
\]

Finally, observing the last equation and introducing the drift vector \( y = [y_1; ...; y_{N_r}]^T \) and the sample vector \( a = [a_1; ...; a_{N_a}]^T \), we see that the drift vector can be written as \( y = F a \) where \( F \) is a \( N_r \times N_a \) matrix such that the \( F_{i,k} = \text{sinc}(\frac{iT_s-kT}{T}) \).

In the general case when there are \( N_t \) > 1 timelines, the development can be repeated for each timeline. In particular, the vectors \( y \) and \( a \) are obtained by stacking the drift and the sample vectors of all the timelines. Moreover, the drift vector can be written as

\[
y = X a \quad \text{(9)}
\]

where \( X \) is a \( D \times K \) block diagonal matrix, with \( K = N_t N_a \), composed by \( N_t \) identical blocks and given by

\[
X = \begin{pmatrix}
F & 0 & 0 & \cdots & 0 \\
0 & F & 0 & \cdots & 0 \\
0 & 0 & F & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & F
\end{pmatrix}.
\]

Eventually, using equations (7) and (9) into equation (6), we can write the PACS data as

\[
d = P m + X a + n
\]

which is our final data model. By comparing the latter expression with equation (3) we see that the PACS data can be cast into the model of equation (3). Therefore, we can exploit the theory developed in the preceding sections. However, we still have to verify that the columns of the \( P \) and \( X \) matrices are linearly independent, as is required in equation (3). Since the matrix \( P \) is controlled by the observation strategy, this can be guaranteed\(^3\) by properly designing the scan. For PACS, this is achieved by providing redundancy and diversity by means of multiple, orthogonal scan lines.

### 4.2 Image formation and ALS implementation

We now consider the image formation problem, which is that of estimating the sky \( m \) knowing the data vector \( d \) and the matrices \( P \) and \( X \). Since the PACS data can be cast into the model of equation (3), we can use the ALS algorithm to compute the LS estimate. Moreover, for this application the ALS can be implemented efficiently, as described in the following.

As a starting point, note that since we are interested only in the image estimation, we can lump together steps 3 and 4 of the ALS and obtain the following, simpler algorithm

0. Set \( \delta = d \). Repeat 1-3 until convergence:

1. \( \hat{m} = (P^T P)^{-1} P^T d \).

2. \( \eta = d - P \hat{m} \).

3. \( \delta = d - P \hat{m} \).

where, in step 3, we introduced the matrix \( \Pi_X = X (X^T X)^{-1} X^T \) which is the projector onto the subspace spanned by the columns of the \( X \) matrix. Then we note that, since \( P \) is sparse and binary, \( P^T P \) is a diagonal matrix, as is easy to check. Therefore, the inversion of step 1 can be performed analytically with a complexity \( O(M) \) and the whole step has a complexity \( O(D) \). Moreover, step 2 can be performed with \( D \) subtractions. Finally, also step 3 can be implemented efficiently. To see how, initially suppose that there is a single timeline. Then, given the structure of the matrix \( F \), it is not difficult to verify that the multiplication \( \Pi_X \eta \) can be implemented by low-pass filtering \( \eta \) with a cut-off frequency of \( f_d/2 \). When there are more timelines, the step can be implemented by separately low-pass filtering each timeline. If the filtering is realised by means of a Fast Fourier Transform (FFT), the whole step has a complexity \( O(D \text{log}_2 D) \).

\(^3\)In practice, a linear dependence will remain, since the constant vector is always in the span of both \( P \) and \( X \). This translates into an unknown offset affecting the image estimate which is easy to compensate. See [4] for a deeper discussion.
5 Results

In order to test the ALS algorithm we developed a simulator of the PACS data, based on equation (6). In order to produce the simulated data, we first produce a synthetic yet realistic sky image, represented by the vector $r$ and shown in figure 1. Next, using the pointing information of a real PACS observation, we sample the sky image to produce the signal vector $s$. Then, the drift vector $y$ is obtained, for each timeline, by low-pass filtering a white Gaussian noise sequence, with a cut-off frequency of $0.001 f_s$. Moreover, the noise vector $n$ is obtained by producing a second white Gaussian noise sequence. Finally, the signal, drift and noise vectors are scaled to match realistic amplitudes and the simulated data vector is computed as $d = s + y + n$.

Before presenting the ALS results, we consider two other estimates. The first one is $\hat{m}$ of equation (5), which is the LS estimate obtained assuming that the drift is absent and that will be called the Simple Least Squares (SLS) estimate. Clearly, since the drift is actually present in the data $d$, such estimate does not produce good images. This is can be verified by observing figure 2, where $\hat{m}$ is plotted and we see that the image is plagued by the drift, which causes the stripes following the scan lines.

As a second approach, we consider removing the drift from the timelines by means of a high-pass filtering. This is an ad-hoc method often exploited in practice, e.g. [12]. In this way an updated data vector is obtained, denoted by $\tilde{d}$, which is, ideally, drift free. Then, we consider the associated linear system $Pm = \tilde{d}$ and solve it in the LS sense. In this way we obtain an estimate that will be called the High-Pass Filter (HPF) estimate and is given by

$$\tilde{m} = (P^T P)^{-1} P^T \tilde{d}.$$  

The HPF estimate for the simulated data is shown in figure 3, when the high-pass filter has a cut-off of $0.002 f_s$. By comparing with figure 2 we see that the impact of the drift has been largely mitigated and the HPF estimate is much better than the SLS one. However, when we filter the timelines to remove the drift, also the sky signal is filtered and the image is distorted. In particular, in figure 3, a chequered pattern following the scan lines can be seen, which is due to the filter ringing around strong sources. Moreover, even if it is difficult to visualise, also low level, diffuse distortion is there.

Finally, we consider the LS image estimate, obtained by using the ALS algorithm with a low-pass
Figure 4: The Alternating Least Squares (ALS) estimate.

<table>
<thead>
<tr>
<th>Estimate</th>
<th>IER (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLS</td>
<td>-3.54</td>
</tr>
<tr>
<td>HPF</td>
<td>13.30</td>
</tr>
<tr>
<td>LS/ALS</td>
<td>15.70</td>
</tr>
</tbody>
</table>

Table 1: Image to Error Ratio (IER) of the various estimates, in dB.

filter with a cut-off frequency of $0.002 f_s$. The corresponding image is plotted in figure 4 and is clearly an improvement with respect to the HPF estimate.

As a last point, we present a quantitative comparison of the estimates. In order to evaluate the quality of an estimate, say $\hat{m}$, we compute the Image to Error Ratio (IER), that is the ratio of the true image variance to the error variance, i.e. $\text{IER} = \frac{\text{var}(r)}{\text{var}(r-\hat{m})}$. The IER is computed similarly for the other estimates and is reported in table 1. Observing the table we see that the ALS can provide more than 2 dB of improvement with respect to the HPF estimate.

6 Conclusion

We introduced and discussed an Alternating Least Squares (ALS) algorithm, which can significantly speed up the solution of some LS problems. We presented an application of the algorithm to an imaging system affected by AWGN and drift, showing that in this set up the ALS is an efficient approach for computing the LS estimate.

Acknowledgements: This work was partially funded by the Sapienza University Research Project C26A1324B9/2013 and by the Italian Space Agency (ASI) within the Hi-GAL project.

References: