VECTOR SPACE SOLUTION TO THE MULTIDIMENSIONAL YULE-WALKER EQUATIONS

Steven M. Kay

University of Rhode Island
Department of Electrical and
Computer Engineering
4 East Alumni Ave., Kingston, RI 02881
(email: kay@ele.uri.edu)

Christopher P. Carbone

Naval Undersea Warfare Center
Division Newport, RI 02841
(email: carbonecp@npt.nuwc.navy.mil)

ABSTRACT

This paper describes a vector space approach to solving the multidimensional (m-D) Yule-Walker equations for an arbitrary region of support. This approach leads to a solution that is simple to implement.

1. INTRODUCTION

Two-dimensional (2-D) autoregressive (AR) modelling has found applications in image processing, sonar, and other areas. 3-D and higher dimensional AR modelling has as yet to be extensively studied but applications to signals that vary in time, frequency, and space can easily be envisioned.

2. PROBLEM STATEMENT

The Yule-Walker equations in one dimension can be very conveniently represented in matrix format (see Kay [1]). The matrix approach works well because the region of support (ROS) for the filter parameters of a 1-D AR process is a line segment, which leads to a set of Yule-Walker equations that can easily be put into a matrix format. This convenient representation leads to relatively simple algorithms that are easily implemented in MATLAB or C.

2-D AR models, on the other hand, can be put into a matrix format but the representation is less natural and more forced. This is caused by the 2-D ROS, which leads to a much more complicated set of linear equations [1],[2]. Less insight is gained from the matrix format and coding is more difficult. Forcing 3-D and higher dimensional models into a matrix format is even more strained and leads to an extremely difficult implementation.

The motivation in using a vector space approach is that it leads to a natural method of representing the m-D Yule-Walker equations. This is because vector spaces easily generalize to higher dimensions. The vector space method also aids one's intuition in developing and coding algorithms.

3. SOLVING THE MULTIDIMENSIONAL YULE-WALKER EQUATIONS

Consider an m-D AR process.

$$x[\mathbf{n}] = -\sum_{\mathbf{k}\in S'_m} a[\mathbf{k}]x[\mathbf{n}-\mathbf{k}] + u[\mathbf{n}] \tag{1}$$

where $\mathbf{n} = [n_1\ n_2\ \ldots\ n_m]^T,\ S'_m$ is the region of support for $a[\mathbf{k}]$, and $u[\mathbf{n}]$ is white noise. The m-D Yule Walker equations are found as follows

$$\begin{aligned}&\mathrm{E}[x[\mathbf{n}]x^*[\mathbf{n}-\mathbf{l}]] = \\ &-\sum_{\mathbf{k}\in S'_m} a[\mathbf{k}]\mathrm{E}[x[\mathbf{n}-\mathbf{k}]x^*[\mathbf{n}-\mathbf{l}]] + \mathrm{E}[u[\mathbf{n}]x^*[\mathbf{n}-\mathbf{l}]].\end{aligned} \tag{2}$$

Defining the autocorrelation function as

$$r_x[\mathbf{k}] = E[x^*[\mathbf{n}]x[\mathbf{n}+\mathbf{k}]]$$

we have from (2) that

$$r_x[\mathbf{l}] = -\sum_{\mathbf{k}\in S'_m} a[\mathbf{k}]r_x[\mathbf{l}-\mathbf{k}] \qquad \mathbf{l}\in S'_m. \tag{3}$$

This assumes that $\mathrm{E}[u[\mathbf{n}]x^*[\mathbf{n}-\mathbf{l}]] = 0$ for $\mathbf{l}\in S'_m$. Note from (1) that $x[\mathbf{n}-\mathbf{l}]$ for $\mathbf{l}\in S'_m$ constitutes the "past", which is uncorrelated with $u[\mathbf{n}]$. Also from (1)

$$\mathrm{E}[|u[\mathbf{n}]|^2] = \mathrm{E}\left[\left|\sum_{\mathbf{k}\in S_m} a[\mathbf{k}]x[\mathbf{n}-\mathbf{k}]\right|^2\right] \tag{4}$$

where $S_m = S'_m \bigcup \{\mathbf{k}=\mathbf{0}\}$ and $a[\mathbf{0}] = 1$. This can be shown to reduce to

$$\begin{aligned}\sigma^2 &= \mathrm{E}\left[\sum_{\mathbf{k}\in S_m} a[\mathbf{k}]x[\mathbf{n}-\mathbf{k}]x^*[\mathbf{n}]\right] \\ &= \sum_{\mathbf{k}\in S_m} a[\mathbf{k}]r_x[-\mathbf{k}].\end{aligned} \tag{5}$$

As a result (3) and (5) can be combined to yield the complete set of Yule-Walker equations

$$\sum_{\mathbf{k}\in S_m} a[\mathbf{k}]r_x[\mathbf{l}-\mathbf{k}] = \sigma^2\delta[\mathbf{l}] \;\text{ for }\; \mathbf{l}\in S_m \tag{6}$$

where

$$\delta[\mathbf{l}] = \begin{cases} 1 & \mathbf{l}=\mathbf{0} \\ 0 & \mathbf{l}\neq\mathbf{0}. \end{cases}$$

Letting $b[\mathbf{k}] = a[\mathbf{k}]/\sigma^2$, this becomes

$$\sum_{\mathbf{k}\in S_m} b[\mathbf{k}]r_x[\mathbf{l}-\mathbf{k}] = \delta[\mathbf{l}] \;\; \mathbf{l}\in S_m. \tag{7}$$

We can view (7) as a linear transformation, which transforms the vector $b[\mathbf{k}]$ into the vector $\delta[\mathbf{k}]$. We represent this transformation symbolically as

$$\mathcal{L}\{b[\mathbf{k}]\} = \delta[\mathbf{k}]. \tag{8}$$

Note that the domain and range spaces are the same since $b[\mathbf{k}]\in\mathcal{V}$, $\delta[\mathbf{k}]\in\mathcal{V}$ where $\mathcal{V}$ is the space of complex sequences defined on S_m.To appreciate the generality of (8) we give a few examples. In Figure 1a we show the usual ROS for the parameters of a 1-D AR process ($a[0]=1$ is included as it is for the following examples). In Figure 1b the ROS is shown for a 2-D quarter plane, in Figure 1c the ROS is shown for a 2-D nonsymmetric half plane and finally, in Figure 1d the ROS is shown for a 3-D nonsymmetric half space. The process parameters are all defined by $b[\mathbf{k}] = a[\mathbf{k}]/\sigma^2$ for $\mathbf{k}\in S_m$.

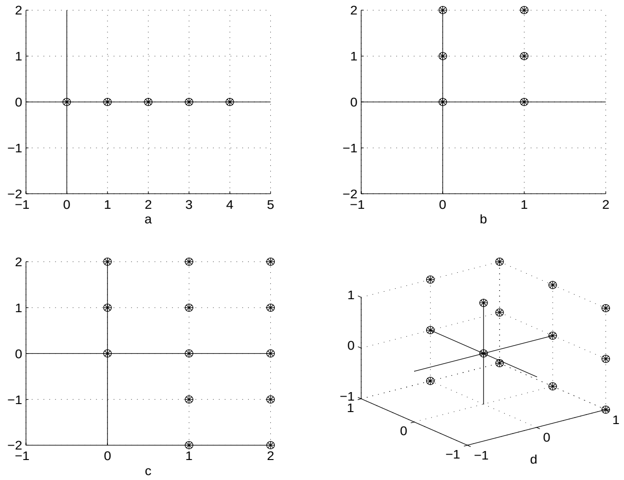

Fig. 1. Illustration of Common Regions of Supports for AR Process Parameters

Now consider v as a *vector* in $\mathcal{V}$. The dimension of $\mathcal{V}$ is equal to the number of elements in S_m and is denoted by $|S_m|$. For example from Figure 1a,b,c,d we have that $|S_1| = 5$, $|S_2| = 6$, $|S_2| = 13$, and $|S_3| = 14$, respectively.

The natural basis for $\mathcal{V}$ is $\{e_1, e_2, \ldots, e_{|S_m|}\}$, where e_i has a 1 in its i^{th} position and 0 otherwise. To solve (8) for $b[\mathbf{k}]$ we first define an inner product as

$$< v, w > = \sum_{\mathbf{k}\in S_m}\sum_{\mathbf{l}\in S_m} v[\mathbf{k}]r_x[\mathbf{l}-\mathbf{k}]w^*[\mathbf{l}]. \tag{9}$$

To verify that this is a valid inner product we note that $r_x[\mathbf{l}-\mathbf{k}]$ can be written as

$$r_x[\mathbf{l}-\mathbf{k}] = E[x^*[\mathbf{k}]x[\mathbf{l}]] \tag{10}$$

and as a result

$$\begin{aligned} < v, w > = & \; E\left[\sum_{\mathbf{k}\in S_m}\sum_{\mathbf{l}\in S_m} v[\mathbf{k}]x^*[\mathbf{k}]x[\mathbf{l}]w^*[\mathbf{l}]\right] \\ = & \; E\left[\sum_{\mathbf{k}\in S_m} v[\mathbf{k}]x^*[\mathbf{k}]\sum_{\mathbf{l}\in S_m} x[\mathbf{l}]w^*[\mathbf{l}]\right] \end{aligned}$$

from which the usual properties of the inner product follow.

Assume we can find an *orthonormal* basis for $\mathcal{V}$ or $\{v_1, v_2, \ldots, v_{|S_m|}\}$ where $< v_i, v_j > = \delta_{ij}$ so that

$$\sum_{\mathbf{k}\in S_m}\sum_{\mathbf{l}\in S_m} v_i[\mathbf{k}]r_x[\mathbf{l}-\mathbf{k}]v_j^*[\mathbf{l}] = \begin{cases} 0 & i\neq j \\ 1 & i=j. \end{cases} \tag{11}$$

Since $b[\mathbf{k}]\in\mathcal{V}$ we can represent it as a linear combination of basis vectors, $b[\mathbf{k}] = \sum_{i=1}^{|S_m|}\beta_i v_i[\mathbf{k}]$ and the Yule-Walker equations then become from (7)

$$\sum_{i=1}^{|S_m|}\beta_i\sum_{\mathbf{k}\in S_m} v_i[\mathbf{k}]r_x[\mathbf{l}-\mathbf{k}] = \delta[\mathbf{l}] \quad \mathbf{l}\in S_m. \tag{12}$$

Now multiply by $v_j^*[\mathbf{l}]$ for each $\mathbf{l}\in S_m$ and sum to yield

$$\sum_{i=1}^{|S_m|}\beta_i\sum_{\mathbf{k}\in S_m}\sum_{\mathbf{l}\in S_m} v_i[\mathbf{k}]r_x[\mathbf{l}-\mathbf{k}]v_j^*[\mathbf{l}] = \sum_{\mathbf{l}\in S_m} v_j^*[\mathbf{l}]\delta[\mathbf{l}] \tag{13}$$

or

$$\sum_{i=1}^{|S_m|}\beta_i < v_i\,,\, v_j > \; = \sum_{\mathbf{l}\in S_m} v_j^*[\mathbf{l}]\delta[\mathbf{l}]. \tag{14}$$

But $< v_i\,,\, v_j > = \delta_{ij}$, so that

$$\beta_j = \sum_{\mathbf{l}\in S_m} v_j^*[\mathbf{l}]\delta[\mathbf{l}] \tag{15}$$

for $j = 1, 2, \ldots, |S_m|$. Therefore,

$$b[\mathbf{k}] = \sum_{i=1}^{|S_m|}\sum_{\mathbf{l}\in S_m} v_i^*[\mathbf{l}]\delta[\mathbf{l}]v_i[\mathbf{k}]. \tag{16}$$

Finally, since $\sum_{\mathbf{l}\in S_m} v_i^*[\mathbf{l}]\delta[\mathbf{l}] = v_i^*[\mathbf{0}]$, the solution becomes

$$b[\mathbf{k}] = \sum_{i=1}^{|S_m|} v_i^*[\mathbf{0}]v_i[\mathbf{k}]. \quad (17)$$

We have assumed that we can find an orthonormal basis for $\mathcal{V}$. To obtain this basis for $\mathcal{V}$ a Gram-Schmidt (GS) orthogonalization is performed on $\{e_1, e_2, \ldots, e_{|S_m|}\}$ which produces $\{v_1, v_2, \ldots, v_{|S_m|}\}$. It proceeds as follows:

$$v_1 = \frac{e_1}{\|e_1\|}$$

and for $k = 2, 3, \ldots, |S_m|$

$$v_k = \frac{e_k - \sum_{i=1}^{k-1} < e_k, v_i > v_i}{\|e_k - \sum_{i=1}^{k-1} < e_k, v_i > v_i\|}. \quad (18)$$

It is important to realize that no matrix inversions are required. The principal source of computation is due to the need to compute inner products. However, these are easily done as follows. The inner products can be computed as

$$\begin{aligned} < v, w > &= \sum_{\mathbf{k}\in S_m}\sum_{\mathbf{l}\in S_m} v[\mathbf{k}]r_x[\mathbf{l}-\mathbf{k}]w^*[\mathbf{l}] \\ &= \sum_{\mathbf{l}\in S_m} w^*[\mathbf{l}]\sum_{\mathbf{k}\in S_m} v[\mathbf{k}]r_x[\mathbf{l}-\mathbf{k}]. \end{aligned}$$

Let $z[\mathbf{l}] = \sum_{\mathbf{k}\in S_m} v[\mathbf{k}]r_x[\mathbf{l}-\mathbf{k}]$. Then,

$$< v, w >= \sum_{\mathbf{l}\in S_m} w^*[\mathbf{l}]z[\mathbf{l}].$$

To calculate the above we shift the r_x array and then perform an element by element array multiply and sum.

4. RESULTS

We will go through two examples to demonstrate some results and suggest how to use MATLAB to solve the Yule-Walker equations. For the first example, we will use the separable 3-D AR(1,1,1) process with QP ROS that was used by Choi (see [3])

$$\begin{aligned} x[n_1,n_2,n_3] &= 0.9x[n_1-1,n_2,n_3] + 0.88x[n_1,n_2-1,n_3] \\ &+ 0.95x[n_1,n_2,n_3-1] - 0.7920x[n_1-1,n_2-1,n_3] \\ &- 0.8550x[n_1-1,n_2,n_3-1] - 0.8360x[n_1,n_2-1,n_3-1] \\ &+ 0.7524x[n_1-1,n_2-1,n_3-1] + u[n_1,n_2,n_3]. \end{aligned}$$

The autocorrelation function is

$$r_x[k_1,k_2,k_3] = 0.9^{|k_1|}0.88^{|k_2|}0.95^{|k_3|}. \quad (19)$$

To use the theory developed in this paper to find the AR parameters the following steps must be coded.

1. Determine the ROS and hence indices of the AR parameters in the ROS.
2. With the indices of the AR parameters determine the correlations needed in $r_x[\mathbf{l}-\mathbf{k}]$ (see 9).
3. Use the Gram-Schmidt process of (18) to find an orthonormal basis using the inner product in (9).
4. The AR parameters are found directly from (17).

Using the above we obtain

$$\begin{aligned} a[0,0,0] &= 1.0000 \\ a[1,0,0] &= -0.9000 \\ a[0,1,0] &= -0.8800 \\ a[0,0,1] &= -0.9500 \\ a[1,1,0] &= 0.7920 \\ a[1,0,1] &= 0.8550 \\ a[0,1,1] &= 0.8360 \\ a[1,1,1] &= -0.7524 \end{aligned}$$

which is the correct result.

For the next example we use another 3-D AR(1,1,1) process but this time it is non-separable and has the NSHP ROS as in Figure 1d.

$$\begin{aligned} x[n_1,n_2,n_3] =& 0.80x[n_1-1,n_2+1,n_3+1] \\ &+ 0.50x[n_1-1,n_2-1,n_3-1] + u[n_1,n_2,n_3]. \end{aligned}$$

The vector space method results in

$$\begin{aligned} a[0,0,0] &= 1.0000 \\ a[1,-1,-1] &= -0.8000 \\ a[1,1,1] &= -0.5000 \end{aligned}$$

the rest of the parameters are zero, as expected.

5. CONCLUSIONS

In this paper we have developed a simple method to solve the m-dimensional Yule-Walker equations with an arbitrary region of support. No matrix inversions are required and coding is easily accomplished. A complete implementation in MATLAB is available upon request.

6. REFERENCES

[1] S. M. KAY, *Modern Spectral Estimation*, Englewood Cliffs, NJ; Prentice-Hall, 1988

[2] D. E. DUDGEON, AND R. M. MERSEREAU *Multidimensional Digital Signal Processing*, Englewood Cliffs, NJ; Prentice-Hall, 1988

[3] B. CHOI, ”An Order-Recursive Algorithm to Solve the 3-D Yule-Walker Equations of Causal 3-D AR Models”, *IEEE Trans. Signal Processing*, Vol. 47, 1999, pp.2491-2502