Computer Science in Economics and Management **4**: 237–259, 1991.

Obtaining Initial Parameter Estimates for Nonlinear Systems using Multicriteria Associative Memories

ROBERT KALABA
Departments of Electrical and Biomedical Engineering, University of Southern California, Los Angeles, CA 90089-1451, U.S.A.

and

LEIGH TESFATSION
Department of Economics and Department of Mathematics, Iowa State University, Ames, IA 50011-1070, U.S.A.

(Received: 11 February 1991; revised: 15 May 1991)

Abstract. Parameter estimation problems for nonlinear systems are typically formulated as nonlinear optimization problems. For such problems, one has the usual difficulty that standard successive approximation schemes require good initial estimates for the parameter vector. This paper develops a simple multicriteria associative memory (MAM) procedure for obtaining useful initial parameter estimates for nonlinear systems. An easily calculated one-parameter family of associative memory matrices is developed; see Equation (25). Each memory matrix is efficient with respect to two criteria: accurate recovery of parameter-output training case associations; and small matrix norm to guard against noise arising from imprecise calculations and observations. For illustration, the MAM procedure is used to obtain initial parameter estimates for a well-known nonlinear economic model, the Solow–Swan growth model. Surprisingly accurate initial parameter estimates are obtained over broad ranges of the family of MAM memory matrices, even when observations are corrupted by i.i.d. or correlated noise.

Key words. Nonlinear estimation, artificial neural networks, associative memory, multicriteria optimization, Solow–Swan growth model.

1. Introduction

Parameter estimation problems for nonlinear systems are typically formulated as nonlinear optimization problems. In statistics and econometrics, the two basic optimization formulations are nonlinear least squares and maximum likelihood.

Numerical methods for solving nonlinear least squares and maximum likelihood problems include iterative methods such as Gauss-Newton and the method of scoring (see, e.g., [3, 6, 20]), and modern simplicial and homotopy continuation methods (e.g., [2, 7, 12, 23]). For quadratically or cubically convergent iterative methods, the components of the initial estimate for the parameter vector often have to be within 10 or 20 percent of their true values (see [4]). Although

continuation methods potentially expand the domain of convergence, as a practical matter it is still desirable to have the initial parameter vector estimate be as accurate as possible.

In a previous study [8], a linear associative memory procedure is proposed for obtaining useful initial parameter estimates for nonlinear systems. In recent years, interest in the construction and study of associative memories for optimally associating one set of items with another has increased dramatically. Associative memory procedures are now used in a wide spectrum of disciplines, including neurophysiology, electrical engineering, computer science, mathematics, and the cognitive sciences. See, for example, the monograph [10] by Kohonen, who largely pioneered the study of linear associative memories, the interesting work on bidirectional associative memories by Kosko [11], and the volumes by Rumelhart and McClelland [19] which provide an accessible introduction to neural networks from a cognitive point of view. In addition, researchers in urban and regional planning have achieved promising experimental results using associative memory techniques to initialize linear programming problems; see [9].

For each parameter vector $\mathbf{r}_i$ in a selected training set $\{\mathbf{r}_1, \ldots, \mathbf{r}_q\}$, the system equations are used to determine a vector $\mathbf{s}_i$ of system outputs. (Both [8] and this paper consider the case in which each parameter vector is associated with a single system output vector. More generally, each parameter vector could determine a nondegenerate probability distribution over system output vectors. In this case, the parameter vector could be associated with a Monte Carlo generated sample of output vectors drawn from this distribution.) A memory matrix $\hat{\mathbf{M}}$ is then constructed which optimally associates each 'stimulus' vector $\mathbf{s}_i$ with its corresponding 'response' vector $\mathbf{r}_i$, in the sense of least squares. Specifically, $\hat{\mathbf{M}} = \mathbf{R}\mathbf{S}^+$, where $\mathbf{R}$ denotes the matrix $[\mathbf{r}_1, \ldots, \mathbf{r}_q]$ of training parameter vectors and $\mathbf{S}^+$ denote the Moore–Penrose generalized inverse of the training output matrix $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_q]$. Given an observed system output vector $\mathbf{s}^*$, an estimate $\hat{\mathbf{r}}$ for the system parameter vector is obtained by setting $\hat{\mathbf{r}} = \hat{\mathbf{M}}\mathbf{s}^*$.

Surprisingly accurate parameter estimates were obtained in [8] for an illustrative nonlinear image processing problem when the observation vectors $\mathbf{s}^*$ were noise free. However, instability problems were encountered when memory matrices constructed from noise-free training vectors were subsequently used to recover parameter estimates from observation vectors corrupted with noise. (Reference [13, p. 701] derives an interesting analytical representation for the association error which results when a memory matrix constructed from noise-free training cases is used to recover information from noisy observations.) As other researchers have noted ([13, 14]), a serious difficulty with the linear associative memory procedure is that the elements of the memory matrix $\hat{\mathbf{M}}$ can have large orders of magnitude relative to the components of the training vectors. This problem can occur if the positive semidefinite matrix $\mathbf{S}\mathbf{S}^T$ has one or more small nonzero eigenvalues. When the problem does occur, the resulting estimates $\hat{\mathbf{r}}$ are highly sensitive to observation noise. Moreover, round-off errors in the computa-

tion of the large elements of $\hat{\mathbf{M}}$ can induce large errors in these estimates even with noise-free observations.

In [8], it was found that adding noise to the training output vectors $\mathbf{s}_i$, or augmenting the training output vectors to include products of the training output levels as in Poggio's suggested nonlinear associative memory procedure [16], substantially improved the accuracy of the resulting parameter estimates. Other linear associative memory studies have proposed using singular value decomposition [13] or probabilistic methods [14] to achieve a better conditioned memory matrix.

This paper develops a simple multicritieria associative memory (MAM) procedure for nonlinear estimation which directly and systematically guards against the ill-conditioning of the memory matrix. Two basic costs are associated with each possible memory matrix $\mathbf{M}$: an *association* cost $C_A(\mathbf{M})$ measuring the extent to which $\mathbf{M}$ fails to associate each training output vector $\mathbf{s}_i$ with its corresponding training parameter vector $\mathbf{r}_i$; and a *size* cost $C_Z(\mathbf{M})$ measuring the extent to which the elements of $\mathbf{M}$ differ from zero. For any given training set $\{(\mathbf{r}_1, \mathbf{s}_1), \ldots, (\mathbf{r}_q, \mathbf{s}_q)\}$, a closed-form easily calculated expression (25) is obtained for a one-parameter family of memory matrices, each memory matrix having the following efficiency property: No other memory matrix achieves lower cost with respect to both the association and the size criteria. Such matrices are referred to as MAM matrices, and their associated cost vectors (C_A, C_Z) are said to constitute the MAM frontier.

The memory matrix $\hat{\mathbf{M}} = \mathbf{R}\mathbf{S}^+$ corresponds to the extreme point of the MAM frontier where the association cost C_A is minimized with no regard for the size cost C_Z, i.e., with no separate consideration given to the size of the elements of $\mathbf{M}$. By moving away from this extreme point along the frontier, bias (training association error) is increased in return for a decrease in the variance of the resulting parameter estimates. The MAM frontier is thus analogous to the coefficient frontier obtained using ridge trace procedures in regression analysis; see, for example, [22]. A key difference, however, is that the purpose of the MAM frontier is to provide one or more useful *initial* parameter estimates for some given nonlinear estimation problem. Consequently, the usual criticism of ridge regression procedures – that they do not necessarily lead to parameter estimators with optimal statistical properties (see, e.g., [6, pp. 915–916]) – is not applicable to MAM.

Section 2 sets out a basic nonlinear dynamic model from economics – the 'Solow–Swan growth model' – used throughout the paper for illustration. Section 3 outlines the possible use of linear associative memories for obtaining parameter estimates for this model. The potential ill-conditioning of the memory matrix $\hat{\mathbf{M}}$ is discussed in Section 4. Section 5 introduces the notion of multicritieria associative memories (MAM). In Section 6, it is shown that the MAM approach can be used to obtain reasonably accurate parameter estimates for the Solow–Swan growth model from observations corrupted either by i.i.d. or correlated noise. Concluding comments are given in Section 7.

2. An Illustrative Economic Growth Problem

Consider an economy which produces a national product $Y(t)$ at each time $t \geq 0$ using capital and labor inputs $K(t)$ and $L(t)$. The production relation for the economy is given by

$$Y(t) = F(K(t), L(t); \theta), \tag{1}$$

where θ is a parameter characterizing the production process, and the production function $F(\cdot; \theta)$: $E^2 \to E$ exhibits constant returns to scale, i.e., for any $(K, L) \geq 0$, $F(cK, cL; \theta) = cF(K, L; \theta)$ for all $c \in [0, 1]$. Denoting time t consumption by $C(t)$ and time t net investment $dK(t)/dt$ by $DK(t)$, and assuming that the amount of capital depreciation at each time t is a constant proportion $\delta \geq 0$ of the existing capital stock $K(t)$, supply equals demand in the time t product market if and only if

$$Y(t) = C(t) + DK(t) + \delta K(t). \tag{2}$$

Time t gross savings $S(t) = Y(t) - C(t)$ are a constant proportion s of time t national product $Y(t)$, where the savings rate s lies between 0 and 1. Thus,

$$C(t) = [1 - s]Y(t). \tag{3}$$

Substituting (3) into the product market clearing condition (2), and rearranging terms, the growth of the capital stock over times $t \geq 0$ is given by

$$DK(t) = sF(K(t), L(t); \theta) - \delta K(t). \tag{4}$$

The labor force $L(t)$ grows at a constant rate $g \geq 0$, with $L(0) > 0$. Let $k(t) = K(t)/L(t)$ and $y(t) = Y(t)/L(t)$ denote the time t capital-labor and income-labor ratios. Using the constant returns to scale assumption for $F(\cdot; \theta)$, the production relation (1) can be expressed in per capita terms as

$$y(t) = F(k(t), 1; \theta) \equiv f(k(t); \theta). \tag{5}$$

Also, the time rate of change of the capital-labor ratio $k(t)$ satisfies

$$Dk(t)/k(t) = DK(t)/K(t) - DL(t)/L(t) = DK(t)/K(t) - g. \tag{6}$$

Finally, define $\lambda = [g + \delta]$, and divide each side of Equation (4) by $L(t)$. Making use of relations (5) and (6), it follows after some manipulation of terms that the time rate of change $Dk(t)$ of the capital-labor ratio $k(t)$ satisfies the differential equation

$$Dk(t) = sf(k(t); \theta) - \lambda k(t), \quad t \geq 0, \tag{7}$$

where the initial capital-labor ratio $k(0)$ is given by some historically determined value $k_0 > 0$. Equation (7) is the basic differential equation for the Solow–Swan growth model, a well-known macroeconomic model which is still very influential. See, for example, [17] and [18, Section 3.2].

At each time t_j, $j = 1, \ldots, m$, with $0 < t_1 < \cdots < t_m$, an observation $k^*(t_j)$ is obtained on the capital-labor ratio $k(t_j)$ in accordance with the measurement relation

$$k^*(t_j) = k(t_j) + n(t_j) . \tag{8}$$

In (8), the term $n(t_j)$ denotes possible noise arising from an imprecise observation procedure. Given the m-dimensional observation vector

$$s^* = \begin{pmatrix} k^*(t_1) \\ k^*(t_2) \\ \\ k^*(t_m) \end{pmatrix}, \tag{9}$$

the problem is to estimate the parameters $(k_0, \theta, s, \lambda)$ which characterize the underlying data-generating process (7).

3. Nonlinear Estimation by Linear Association

Let the solution to the basic Solow–Swan differential equation (7) be denoted by

$$k(t) = H(t; k_0, \theta, s, \lambda) , \quad t \geq 0 . \tag{10}$$

A classical approach to the parameter estimation problem for the Solow–Swan growth model would be to pose it as a nonlinear least squares problem in which the sum of squared deviations

$$\sum_{j=1}^{m} [k^*(t_j) - H(t_j; k_0, \theta, s, \lambda)]^2 \tag{11}$$

is to be minimized with respect to $(k_0, \theta, s, \lambda)$. For each different trajectory of observations $(k^*(t_1), \ldots, k^*(t_m))$, a different sum (11) would have to be minimized, typically by means of a successive approximation scheme [20]. A major drawback of many successive approximation schemes, however, is the need to have a good initial estimate for the true parameter vector.

A simple linear associative memory procedure will now be presented for obtaining initial parameter estimates for the solution of nonlinear least squares problems such as (11). The basic idea is to find an inverse linear mapping $\hat{\mathbf{M}}$ from the system outputs to the parameters using a supervised training procedure. The training procedure exploits the fact that it is generally easier to go from a parameter vector to an associated system output vector than from a system output vector to an associated parameter vector.

For example, when the system equations are given in the form of a differential system such as (7), an output vector (i.e., a solution trajectory) can be obtained for any specified parameter vector by either analytical or numerical integration. In the case of a reduced form system of equations $q = F(r)$, an output vector q is obtained for any given parameter vector r by the direct numerical evaluation of

$F(r)$. Alternatively, associated pairs of parameter vectors and output vectors might be directly culled from empirical studies or from laboratory experiments. In the latter two cases, the actual underlying system equations would not have to be known.

The first step in the linear associative memory procedure is the construction of a finite set of training cases,

$$\mathbf{r}_i = \begin{pmatrix} k_0 \\ \theta \\ s \\ \lambda \end{pmatrix}_i , \qquad s_i = \begin{pmatrix} k(t_1) \\ k(t_2) \\ \\ k(t_m) \end{pmatrix}_i , \quad i = 1, 2, \ldots, q , \tag{12}$$

where q is the number of training cases, $\mathbf{r}_i$ is a potential vector of system parameters, and the jth component of $\mathbf{s}_i$ is the time-t_j system output $k(t_j) = H(t_j; \mathbf{r}_i)$ obtained by the analytical or numerical integration of the basic Solow–Swan differential equation (7) with $\mathbf{r} = \mathbf{r}_i$ over the time interval $[0, t_m]$. The training cases (12) are used to form the response and stimulus matrices

$$\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2 \cdots \mathbf{r}_q)_{4\times q} , \qquad \mathbf{S} = (s_1, s_2 \cdots s_q)_{m\times q} . \tag{13}$$

An optimal (least-squares) linear mapping between the training stimulus and response matrices (13) is determined by solving

$$\min_M \|\mathbf{MS} - \mathbf{R}\|^2 , \tag{14}$$

where $\|\cdot\|$ denotes the usual Euclidean matrix norm. (The Euclidean matrix norm of any matrix $\mathbf{A}$ is $\|\mathbf{A}\| = \sqrt{\mathrm{tr}(\mathbf{A}^T\mathbf{A})}$.) As originally established in [15], among all possible solutions to problem (14), the solution with minimum norm takes the form

$$\hat{\mathbf{M}} = \mathbf{RS}^+ , \tag{15}$$

where $\mathbf{S}^+$ denotes the Moore–Penrose generalized inverse* of $\mathbf{S}$. Following Kohonen [10, pp. 53–54], the 'best' solution for problem (14) is taken to be the minimum-norm solution (15).

Once the memory matrix $\hat{\mathbf{M}}$ is constructed using a finite set of training cases, it can be used to generate an estimate for the actual vector of system parameters on the basis of actual system output observations. Specifically, given any observed system output vector $\mathbf{s}^*$ for the Solow–Swan growth model, with $\mathbf{s}^*$ taking the form (9), an estimate $\hat{\mathbf{r}}$ for the parameter vector $\mathbf{r} = (k_0, \theta, s, \lambda)^T$ is determined by

$$\hat{\mathbf{r}} = \hat{\mathbf{M}}\mathbf{s}^* . \tag{16}$$

As long as the training cases used to construct the associative memory matrix $\hat{\mathbf{M}}$ remain relevant, new parameter estimates $\hat{\mathbf{r}}$ can be generated for new $m \times 1$

* Given any matrix $\mathbf{A}$, the Moore–Penrose generalized inverse of $\mathbf{A}$ is the unique matrix $\mathbf{A}^+$ satisfying the following four conditions: (i) $\mathbf{AA}^+\mathbf{A} = \mathbf{A}$; (ii) $\mathbf{A}^+\mathbf{AA}^+ = \mathbf{A}^+$; (iii) $(\mathbf{A}^+\mathbf{A})^T = \mathbf{A}^+\mathbf{A}$; and (iv) $(\mathbf{AA}^+)^T = \mathbf{AA}^+$.

observation vectors $\mathbf{s}^*$ by repeated application of the simple matrix operation (16).

It may seem incredible that a *linear* operation such (16) could yield reasonably accurate parameter estimates for a highly *nonlinear* system such as (7). The crucial point is that the memory matrix $\hat{\mathbf{M}}$ encodes the nonlinear inverse mapping between the output vector $\mathbf{s}_i$ and the parameter vector $\mathbf{r}_i$ for each parameter vector $\mathbf{r}_i$ in the selected training set $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_q\}$. The training procedure can be compared with the maximum likelihood method for associating the most 'likely' parameter vector with any given output.

Nevertheless, a serious practical difficulty with the proposed procedure (16) is that the memory matrix $\hat{\mathbf{M}}$ can have large elements, leading to instability in the resulting parameter estimates. The following section focuses on this issue.

4. Difficulties with $\hat{M} = RS^+$

Two potentially important sources of noise arise in linear associative memory procedures. First, due to round-off and truncation errors, computer calculations in both the training stage and the estimation stage are always carried out with some degree of imprecision. Second, extraneous noise may enter into the determination of the observation vectors $\mathbf{s}^*$ in the estimation stage. Both types of noise can induce large errors in the resulting parameter estimates when the elements of the memory matrix $\hat{\mathbf{M}}$ have large orders of magnitude relative to the components of the training vectors.

Suppose, for example, that the calculation of the memory matrix $\hat{\mathbf{M}}$ is corrupted by a noise matrix $\mathbf{N}$, so that $\hat{\mathbf{M}} = \mathbf{M}^0 + \mathbf{N}$. Let $\mathbf{s}$ be a system output vector corresponding to a parameter vector $\mathbf{r}$ in the span of the training parameter set, so that the components of $\mathbf{s}$ and $\mathbf{r}$ have about the same orders of magnitude as the components of the training output vectors and the training parameter vectors, respectively. The parameter estimate associated with $\mathbf{s}$ by $\hat{\mathbf{M}}$ in accordance with (16) then takes the form

$$\hat{\mathbf{r}} = \hat{\mathbf{r}}^0 + \hat{\mathbf{r}}^N = \mathbf{M}^0\mathbf{s} + \mathbf{N}\mathbf{s} = \hat{\mathbf{M}}\mathbf{s}\,. \tag{17}$$

If the elements of the noise matrix $\mathbf{N}$ have orders of magnitude which are about the same as the elements of $\mathbf{M}^0$, and the latter elements have large orders of magnitude relative to the components of the training vectors, then the parameter estimate noise vector $\mathbf{r}^N = \mathbf{N}\mathbf{s}$ can be large relative to the parameter estimate signal vector $\mathbf{r}^0 = \mathbf{M}^0\mathbf{s} \approx \mathbf{r}$.

Alternatively, let $\mathbf{s}$ again denote a system output vector corresponding to a parameter vector $\mathbf{r}$ in the span of the training parameter set; but suppose this time that the memory matrix $\hat{\mathbf{M}}$ has been accurately calculated, and $\hat{\mathbf{M}}\mathbf{s}$ is close to $\mathbf{r}$. Let $\mathbf{n}$ denote a noise vector which corrupts the observation of $\mathbf{s}$, so that the observed output vector takes the noisy form $\mathbf{s}^* = [\mathbf{s} + \mathbf{n}]$. The parameter estimate generated as in (16) is then

$$\hat{\mathbf{r}} = \hat{\mathbf{r}}^s + \hat{\mathbf{r}}^n = \hat{\mathbf{M}}\mathbf{s} + \hat{\mathbf{M}}\mathbf{n} = \hat{\mathbf{M}}\mathbf{s}^* \,. \tag{18}$$

If the components of the noise vector $\mathbf{n}$ have orders of magnitude which are about the same as the components of $\mathbf{s}$, and the elements of $\hat{\mathbf{M}}$ have large orders of magnitude relative to the training vectors, then multiplying the noisy observation vector $\mathbf{s}^*$ by $\hat{\mathbf{M}}$ can result in a parameter estimate noise vector $\hat{\mathbf{r}}^n = \hat{\mathbf{M}}\mathbf{n}$ which is large relative to the parameter estimate signal vector $\hat{\mathbf{r}}^s = \hat{\mathbf{M}}\mathbf{s} \approx \mathbf{r}$.

Consequently, once the possibility of imprecise calculations and observations is recognized, keeping the magnitudes of the elements of the memory matrix small becomes an important criterion in addition to the basic criterion of obtaining good training case associations. This observation is of course just a special case of a long-recognized point in linear estimation theory – the desirability of reducing the norm of a linear estimator in order to enhance the numerical stability of the resulting estimates.

It is difficult to imagine that machine-dependent calculation-induced noise obeys any meaningful probability relations. Consequently, it seems difficult to deduce interesting general relationships between this noise and the 'optimal' size of the memory matrix. However, more can be said in the case of observation noise.

Let $\mathbf{S}^n = \mathbf{S} + \mathbf{N}$ denote an $m \times q$ training stimulus matrix which is the sum of a given matrix $\mathbf{S}$ of noise-free training stimulus vectors and a given matrix $\mathbf{N}$ of noise elements. Also, let $\mathbf{R}$ denote a given $n \times q$ matrix of training response vectors. The usual criterion function for the choice of the memory matrix $\mathbf{M}$ can then be written in the expanded form

$$\begin{aligned} \|\mathbf{MS}^n - \mathbf{R}\|^2 &\equiv \text{Trace}\,([\mathbf{MS}^n - \mathbf{R}]^T[\mathbf{MS}^n - \mathbf{R}]) \\ &= \text{Trace}\,([\mathbf{M}(\mathbf{S}+\mathbf{N}) - \mathbf{R}]^T[\mathbf{M}(\mathbf{S}+\mathbf{N}) - \mathbf{R}]) \\ &= \|\mathbf{MS} - \mathbf{R}\|^2 + \|\mathbf{MN}\|^2 + 2\,\text{Trace}\,([\mathbf{MS} - \mathbf{R}]^T - \mathbf{MN})\,. \end{aligned} \tag{19}$$

In each experiment conducted in [8] using a noisy training stimulus matrix $\mathbf{S}^n$, the memory matrix $\hat{\mathbf{M}}^n$ which minimized the criterion function (19) had elements of smaller magnitude (and subsequently yielded more precise parameter estimates) than the corresponding memory matrix $\hat{\mathbf{M}}$ which minimized a modified noise-free form of the criterion function (19) with $\mathbf{N}$ replaced by a matrix of zeros.

One possible explanation for these experimental findings is that adding noise to the rows of the $m \times q$ training stimulus matrix $\mathbf{S}$ will generally destroy any linear dependencies or near-dependencies among these rows. This improves the conditioning of the $m \times m$ matrix $\mathbf{SS}^T$, which is turn tends to reduce the size of the elements of the memory matrix $\hat{\mathbf{M}} = \mathbf{RS}^+$. This statement is a corollary of the following well-known representation for the Moore–Penrose generalized inverse of any matrix $\mathbf{A}$: $\mathbf{A}^+ = p_1^{-1/2} v_1 w_1^T + \cdots + p_r^{-1/2} v_r w_r^T$, where r is the rank of $\mathbf{A}$, the scalars p_i, $i = 1, \ldots, r$, are the nonzero eigenvalues of $\mathbf{A}^T\mathbf{A}$ (and of $\mathbf{AA}^T$) repeated according to their multiplicity, and the column vectors v_i and w_i, $i = 1, \ldots, r$, are the corresponding orthonormal eigenvectors of $\mathbf{A}^T\mathbf{A}$ and of $\mathbf{AA}^T$, respectively. See,

for example, [1]. However, additional insight regarding these experimental results is obtained by considering the average form of the criterion function (19) across experiments.

The elements n_{jk} of the noise matrix **N** in (19) are given numbers representing realized perturbations in the system observations for the particular experiment under consideration. No assumptions are made concerning the nature of these perturbations. Suppose instead that (19) is interpreted as an *ex ante* criterion function in which the elements of **N** are yet-to-be realized i.i.d. random variables with mean zero and variance σ^2. Suppose also that the memory matrix **M** is chosen to minimize the expected value of the criterion function (19). That is, suppose a mean-square error (mse) criterion function is used to select **M**.

To determine the form which this mse criterion function takes, first consider a preliminary technical observation. Let $\mathbf{U} \equiv \mathbf{MN}$, where **M** is any given $n \times m$ matrix, and let u_{ik} denote the ith component of the kth column of **U**. Then u_{ik} has zero mean and a variance σ_{ik}^2 given by

$$\sigma_{ik}^2 = \mathbf{E}\left\{\sum_j (M_{ij} \cdot n_{jk})^2\right\} = \sigma^2 \sum_j M_{ij}^2 . \tag{20}$$

The variance of u_{ik} is therefore proportional to the sum of the squares of the entries of **M** in the ith row, and this variance does not depend on k.

Using (20), it can be shown that the mse criterion function with i.i.d. mean-zero observation noise takes the form

$$E\{\text{Trace}\,([\mathbf{M}(\mathbf{S}+\mathbf{N})-\mathbf{R}]^T[\mathbf{M}(\mathbf{S}+\mathbf{N})-\mathbf{R}])\} = \|\mathbf{MS}-\mathbf{R}\|^2 + q\sigma^2\|\mathbf{M}\|^2 . \tag{21}$$

Note that $\|\mathbf{M}\|^2$, i.e., the sum of the squares of the elements of **M**, enters additively into the right-hand side of (21) with a weight factor $q\sigma^2$. Consequently, (21) reveals that the mse criterion function with i.i.d. mean-zero observation noise is a penalty function which takes into account two different criteria for the choice of **M**: namely, good training case associations (i.e., $\mathbf{MS} \approx \mathbf{R}$); and small-sized elements. One would therefore expect the elements of the matrix **M** which minimizes (21) for the noise case $\sigma^2 > 0$ to be reduced in magnitude in comparison with the noise-free case $\sigma^2 = 0$.

In view of these observations, why not just use a mse criterion for the choice of the memory matrix? One difficulty is that the resulting 'optimal' memory matrix depends on the first and second moments of the noise terms, and these moments may be unknown or even non-existent. A second difficulty is that only *observation* noise is controlled for; in the absence of observation noise, no attention is paid to the size of the elements of the memory matrix. As previously noted, however, the noise resulting from round-off and truncation errors in the calculation of the memory matrix can be significant when the memory matrix elements are large due to the ill-conditioning of $\mathbf{SS}^T$. Experiments demonstrating this difficulty are reported in Section 6 (Table I).

In the following section an alternative multicriteria associative memory (MAM) procedure is developed which directly guards against the ill-conditioning of the memory matrix. Although motivated by the penalty function representation (21) for the mse criterion function with i.i.d. observation noise, the MAM procedure does not require a priori knowledge of the observation noise statistics; and calculation noise is guarded against as well as observation noise.

5. Multicriteria Associative Memory

5.1. THE BASIC MAM APPROACH

Once the possibility of imprecise calculations and observations is recognized, keeping the elements of the memory matrix small becomes an important criterion in addition to the basic criterion of obtaining good training case associations. Consequently, given an $n \times q$ training response matrix $\mathbf{R}$ and an $m \times q$ training stimulus matrix $\mathbf{S}$, two basic costs are incurred by the choice of an $n \times m$ memory matrix $\mathbf{M}$: an *association* cost $C_A(\mathbf{M})$ measuring the extent to which $\mathbf{M}$ fails to associate the training stimulus matrix $\mathbf{S}$ with the training response matrix $\mathbf{R}$; and a *size* cost $C_Z(\mathbf{M})$ measuring the extent to which the elements of $\mathbf{M}$ differ from zero.

On the basis of both tractability and general intuitive appeal, the costs $C_A(\mathbf{M})$ and $C_Z(\mathbf{M})$ are each expressed as sums of squared discrepancy terms. Specifically, the association cost entailed by $\mathbf{M}$ is taken to be

$$C_A(\mathbf{M}) = \|\mathbf{MS} - \mathbf{R}\|^2 , \tag{22}$$

and the size cost entailed by $\mathbf{M}$ is taken to be

$$C_Z(\mathbf{M}) = \|\mathbf{M}\|^2 . \tag{23}$$

The memory matrices of interest are those which are cost-efficient with respect to the two-dimensional cost vectors (C_A, C_Z). Call a memory matrix $\mathbf{M}$ 'cost-subordinated' if there exists another memory matrix $\mathbf{M}^*$ which yields a lower value for one type of cost without increasing the value of the other. Memory matrices which are *not* cost-subordinated are referred to as MAM matrices.

The cost vectors (C_A, C_Z) entailed by the MAM matrices are said to constitute the MAM frontier. By construction, the MAM frontier is a downward sloping strictly convex curve in the two-dimensional (C_A, C_Z) cost plane (see Figure 1). At one extreme of the MAM frontier is the cost vector incurred when the association cost $C_A(\mathbf{M})$ is minimized with no regard for the size cost $C_Z(\mathbf{M})$. One memory matrix which minimizes the association cost is $\hat{\mathbf{M}} = \mathbf{RS}^+$, the memory matrix determined using the standard linear associative memory approach outlined in Section 3. If the rank of the $m \times q$ matrix $\mathbf{S}$ is q, the minimum association cost is zero. Otherwise, the minimum association cost will generally be positive. In any case, among all memory matrices which achieve the minimum association

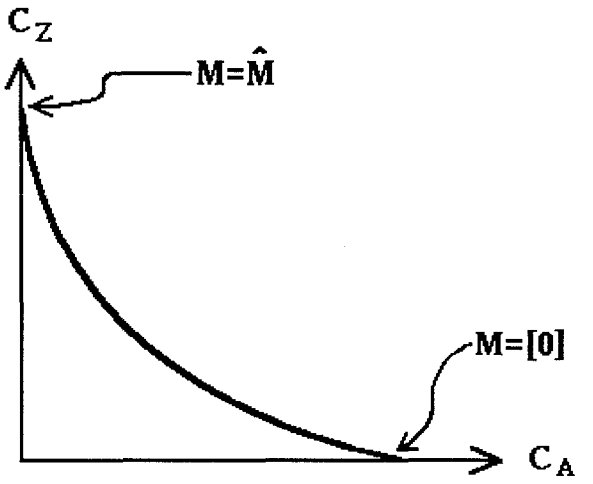

Fig.1: The MAM Frontier

cost, the memory matrix $\hat{\mathbf{M}}$ is the one which has the smallest norm. See [10, pp. 48–54]. At the other extreme of the MAM frontier is the cost vector incurred when the size cost $C_Z(\mathbf{M})$ is minimized with no regard for the association cost $C_A(\mathbf{M})$. The memory matrix which uniquely solves this minimization problem is the zero matrix [0].

5.2. PARAMETRIC REPRESENTATION FOR THE MAM FRONTIER

In view of the strict convexity of the MAM frontier, each point on this frontier solves a problem of the form 'minimize C_Z subject to C_A = constant'. Consequently, each MAM matrix can be generated as the solution to a problem of the form

$$\min_{\mathbf{M}}[\alpha C_A(\mathbf{M}) + [1-\alpha]C_Z(\mathbf{M})]\,, \tag{24}$$

where α is a suitably chosen Lagrange multiplier lying between 0 and 1. The slope of the MAM frontier at the solution point for (24) is given by $-(\alpha/[1-\alpha])$. Thus, α parameterizes the attainable trade-offs between association cost and size cost along the MAM frontier.

Making repeated use of trace derivative formulas,* one sees that the solution to problem (24) is given by

$$\hat{\mathbf{M}}(\alpha) = \alpha\mathbf{R}\mathbf{S}^T(\alpha\mathbf{S}\mathbf{S}^T + [1-\alpha]\mathbf{I})^{-1}\,. \tag{25}$$

Calculation of the MAM matrix (25) is straightforward; only an ordinary inverse operation is needed. As is well known (see, e.g., [1]),

* These trace derivative formulas are as follows: For any conformable matrices **A** and **B** with a square product **AB**, $\partial\,\mathrm{Trace}(\mathbf{AB})/\partial\mathbf{A} = \mathbf{B}^T$; and, for any conformable matrices **A** and **B** with **B** symmetric, $\partial\,\mathrm{Trace}\,\mathbf{A}\mathbf{B}\mathbf{A}^T/\partial\mathbf{A} = 2\mathbf{A}\mathbf{B}^T$.

$$\mathbf{S}^{+} = \lim_{\beta \to 0} [\mathbf{S}^{T}(\mathbf{S}\mathbf{S}^{T} + \beta^{2}\mathbf{I})^{-1}] . \tag{26}$$

It follows from (25) and (26) that the MAM matrix $\hat{\mathbf{M}}(\alpha)$ converges to the Kohonen memory matrix $\hat{\mathbf{M}} = \mathbf{R}\mathbf{S}^{+}$ as $\alpha \in (0, 1)$ converges to 1 and to the zero matrix [0] as $\alpha \in (0, 1)$ converges to 0. (See Figure 1.) Consequently, the one-parameter family of MAM matrices of form (25) is a generalization of the usual memory matrix construction. By varying the parameter α, the norm of the memory matrix can be flexibly controlled in a trade-off against association error.

5.3. MAM ESTIMATES

Let $\mathbf{s}^{*}$ be any given $m \times 1$ observation vector. For each α in $[0, 1]$, the parameter vector estimate corresponding to the MAM matrix $\hat{\mathbf{M}}(\alpha)$ is

$$\hat{\mathbf{r}}(\alpha) = \hat{\mathbf{M}}(\alpha)\mathbf{s}^{*} . \tag{27}$$

The estimate (27) is referred to as a MAM estimate.

Given any nonlinear least squares problem such as (11), the MAM procedure can be used to generate a range of MAM estimates $\hat{\mathbf{r}}(\alpha)$ for the underlying system parameter vector. The weight factor α is a tuning device which can be adjusted up or down to control for noise in the observation vectors as well as for noise due to round-off and truncation errors. The objective is to determine, through the training process, a range of values for α which result in one or more usable initial parameter estimates for the solution of the nonlinear least squares problem by a successive approximation scheme.

The following section reports on a variety of numerical experiments undertaken to see whether the MAM procedure delivers usable initial parameter estimates for the Solow–Swan growth model.

6. Numerical Examples

The first step in the MAM procedure is the construction of a finite set of training cases. As noted in Section 3, given any training parameter vector $\mathbf{r}_i = (k_0, \theta, s, \lambda)^{T}$ for the Solow–Swan growth model, a corresponding training output vector $\mathbf{s}_i = (k(t_1), \ldots, k(t_m))^{T}$ can be generated by numerically integrating the basic Solow–Swan differential equation (7). A closed-form expression for the solution of this differential equation is not required.

For conceptual clarity, however, it is useful to focus on a special case in which a closed-form expression for the solution of (7) can be obtained. Specifically, suppose the per-capita production function $f(\cdot; \theta)$ for the Solow–Swan growth model takes the commonly used Cobb–Douglas form

$$f(k; \theta) = k^{\theta} \tag{28}$$

for some $\theta \in (0, 1)$. The production parameter θ then gives the capital share of

the national product at each time t. That is, $\theta = p(t)k(t)/y(t)$, where the time t capital rental rate $p(t)$ is taken to be the time t marginal product of capital $f'(k(t))$.

Given (28), the solution to the basic Solow–Swan differential equation (7) is*

$$k(t) = H(t; k_0, \theta, s, \lambda)$$
$$= ([(k_0)^{1-\theta} - s/\lambda]\, e^{-[1-\theta]\lambda t} + s/\lambda)^{1/[1-\theta]}, \quad t \geq 0. \tag{29}$$

Note that (29) is a highly nonlinear function of the four model parameters k_0, θ, s, and λ. For each given parameter vector $(k_0, \theta, s, \lambda)$, the solution value (29) for the time t capital-labor ratio $k(t)$ converges as t approaches infinity to the stationary solution value

$$\bar{k} = \left(\frac{s}{\lambda}\right)^{1/(1-\theta)} \tag{30}$$

6.1. GENERAL EXPERIMENTAL SET-UP

The MAM procedure can be implemented even if nothing is known a priori about the nonlinear least squares solution for the parameter vector. However, if prior information is available concerning a plausible range of values for this solution, the training parameter set should presumably be designed to encompass this range.

To demonstrate the MAM procedure for the Solow–Swan growth model over a plausible training grid, the training parameter sets $\{\mathbf{r}_1, \ldots, \mathbf{r}_q\}$ for all of the numerical experiments reported below were constructed on the basis of the following guideline parameter values:

$$k_0 \approx 5.0; \quad \theta \approx 0.29; \quad s \approx 0.15; \text{ and } \lambda = [g + \delta] \approx [0.03 + 0.07] = 0.10. \tag{31}$$

The guideline values for the capital share θ, the gross savings rate s, the effective labor growth rate g, and the depreciation rate δ in (31) were constructed using empirically determined ratios and magnitudes given in [5] and [21]. As noted in [21, p. 150, footnote 1], the precise meaning of the 'capital stock' $K(t)$ and, hence, the capital-labor ratio $k(t)$, is a source of much controversy in growth theory. The guideline value for k_0 in (31) is for illustrative purposes only.

To simplify graphical depictions and comparisons, the training parameter set for each experiment consisted of 49 two-dimensional parameter vectors $\mathbf{r}_i$, with

* Given (28) and $k_0 > 0$, it follows from a simple phase diagram argument that $k(t) > 0$ for all $t \geq 0$ along the solution path for (7). Introduce the variable transformation $u(t) = k(t)/f(k(t)) = k(t)^{1-\theta}$, so that $Du(t)/u(t) = [1 - \theta]Dk(t)/k(t)$. In terms of $u(t)$, Equation (7) reduces to a nonhomogeneous linear differential equation with constant coefficients. The solution to this transformed differential equation is $u(t) = [c \exp(-[1 - \theta]\lambda t) + s/\lambda]$, where $c = [u(0) - s/\lambda]$. The solution (29) for $k(t)$ is then easily obtained by an inverse transformation.

constant guideline values set for the remaining two parameters. One series of experiments ('Series I') was run for training parameter vectors of the form $\mathbf{r}_i = (k_0, \theta)^T$, and another series ('Series II') was run for training parameter vectors of the form $\mathbf{r}_i = (s, \theta)^T$.

In each experiment, seven different values were considered for each of the two training parameters, and these values were centered around the parameters' guideline values in (31). Consequently, in each experiment the 'training parameter grid' consisted of a 7×7 square of two-dimensional parameter points approximately centered at a guideline point as determined from (31).

The experiments reported below use the following grid mesh specifications. For the Series I experiments with 2×1 training parameter vectors of the form $\mathbf{r}_i = (k_0, \theta)^T$, the values considered for k_0 were $\{4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0\}$ and the values considered for θ were $\{0.20, 0.23, 0.26, 0.29, 0.32, 0.35, 0.38\}$; and the remaining two parameters s and λ were kept constant at the guideline values $s = 0.15$ and $\lambda = 0.10$. For the Series II experiments with 2×1 training parameter vectors of the form $\mathbf{r}_i = (s, \theta)^T$, the values considered for s were $\{0.09, 0.11, 0.13, 0.15, 0.17, 0.19, 0.21\}$ and the values considered for θ were as listed above; and the remaining two parameters k_0 and λ were kept constant at the guideline values $k_0 = 5.0$ and $\lambda = 0.10$.

Each experiment in each series consisted of four basic steps.

- First, for each considered training parameter set $\{\mathbf{r}_1, \ldots, \mathbf{r}_{49}\}$, a corresponding training output set $\{\mathbf{s}_1, \ldots, \mathbf{s}_{49}\}$ was generated using the closed form solution (29). Each 15×1 training output vector $\mathbf{s}_i$ consisted of 15 capital-labor ratios $k(t_j)$, $j = 1, \ldots, 15$, calculated for the 'observation times' $t_1 = 0.05$, $t_2 = 1.05, \ldots, t_{15} = 14.05$. (Surprisingly, comparable levels of accuracy were also obtained for the MAM estimates when the same time interval $[0.05, 14.05]$ was spanned by only five or ten observation times t_j, or even when only five or ten observation times t_j were used with a step size of 1.00 to span the smaller time intervals $[0.05, 4.05]$ and $[0.05, 9.05]$, respectively.) These training vectors were used to form training response and stimulus matrices, as in (13).
- Second, 2×15 MAM matrices $\hat{\mathbf{M}}(\alpha)$ were constructed as in (25), with α ranging from 0.10 to 1.00.
- Third, the components of the 15×1 training output vectors $\mathbf{s}_i$ were corrupted with additive noise, resulting in a set of noisy 'observation vectors' of the form $\mathbf{s}_i^* = \mathbf{s}_i + \mathbf{n}_i$.
- Fourth, tests were conducted to determine the extent to which each 2×15 MAM matrix $\hat{\mathbf{M}}(\alpha)$ was successfully able to recover the 2×1 training parameter vector $\mathbf{r}_i$ when postmultiplied by the 15×1 noisy observation vector $\mathbf{s}_i^*$, $i = 1, \ldots, 49$. For each training case i, the measure used to judge the success of the recovery was the discrepancy, in percentage terms, between the jth component $\hat{\mathbf{r}}_{ij}(\alpha)$ of the MAM estimate $\hat{\mathbf{r}}_i(\alpha) = \hat{\mathbf{M}}(\alpha)\mathbf{s}_i^*$ and

the jth component $\mathbf{r}_{ij}$ of the actual training parameter vector $\mathbf{r}_i$, $j = 1, 2$:

$$\text{dis}_{ij}(\alpha) = ([\hat{\mathbf{r}}_{ij}(\alpha) - \mathbf{r}_{ij}]/\mathbf{r}_{ij}) \times 100 . \tag{32}$$

In each series of experiments, two issues were investigated: (1) the effect of i.i.d. observation noise on the accuracy of the resulting parameter estimates; and (2) the effect of correlated observation noise on the accuracy of the resulting parameter estimates.

6.2. EXPERIMENTAL RESULTS WITH i.i.d. OBSERVATION NOISE

Suppose the observation vectors $\mathbf{s}_i^*$ are taken to be the training output vectors $\mathbf{s}_i$ corrupted by additive i.i.d. noise with mean 0 and variance σ^2. In this case, to achieve the minimum mse recovery of the training parameter vectors over the training grid, it follows from (21) that the penalty weight α in (24) should be set equal to

$$\alpha^0 = 1/[1 + q\sigma^2] . \tag{33}$$

In particular, for any given number of training cases q, the penalty weight α should be set close to 1.00 when observation noise is minimal ($\sigma^2 \approx 0$) and close to 0 when observation noise is extensive ($\sigma^2 \gg 0$).

Note that α^0 is independent of the design of the training output matrix $\mathbf{S}$ apart from its row dimension q. Although α^0 varies in response to changes in the variance σ^2 of the observation noise, it does not vary in response to changes in the conditioning of $\mathbf{SS}^T$. Thus, a second potentially important source of noise arising in linear associative memory procedures is not taken into account: namely, round-off and truncation errors in the calculation of the elements of the memory matrix $\hat{\mathbf{M}}$ when these elements are large due to the ill-conditioning of $\mathbf{SS}^T$. As will be seen below (Table I), the experimentally determined optimal values for α in experiments with noise-free observation vectors were always below the theoretically determined minimum mse value $\alpha^0 = 1.00$, suggesting that the matrices $\mathbf{SS}^T$ generated in these experiments were ill-conditioned.

A variety of numerical experiments were run for the Solow–Swan growth model with observation vectors $\mathbf{s}_i^*$ taken to be the training output vectors $\mathbf{s}_i$ with components corrupted by additive i.i.d. noise generated by means of a $N(0, \sigma^2)$ pseudorandom number generator. Recoverability was investigated for standard deviation levels σ ranging from 0 to 0.40, where, in each experiment, $\sigma = 0.\ zx$ represented a measurement error of about $zx/2$ percent. (For the guideline values (31), the stationary limiting solution (30) for the capital-labor ratio $k(t)$ is about 1.8, so that the typical value of $k(t)$ along the solution trajectory is about 2.0.)

The MAM estimates $\hat{\mathbf{r}}_i(\alpha)$ determined in these experiments were reasonably accurate when α was set equal to a value in [0.10, 0.99] which was roughly nearby the minimum mse value (33), e.g., a distance apart of 0.20 or less. Overall, high accuracy levels (discrepancies around one percent or less) were attained for the

initial capital-labor ratio k_0, good accuracy levels (discrepancies around 10 percent or less) were obtained for the gross savings rate s, and reasonably good accuracy levels (discrepancies around twenty percent or less) were obtained for the capital share θ. Moreover, recovery accuracy was generally good for all training parameter vectors lying near the center of the training parameter grids. In all of these experiments, the elements of the MAM matrices $\hat{\mathbf{M}}(\alpha)$ had orders of magnitude ranging from about 10^{-2} to 10^{0}.

On the other hand, when α was set equal to a value very near 1.00, the elements of the MAM matrix increased in size by several orders of magnitude. For example, when α was set equal to 0.9999999999, the elements were typically on the order of 10^3; and when α was set equal to 1.00, the elements were typically on the order of 10^4 or 10^5. Not surprisingly, the parameter estimates generated by these MAM matrices were not robust with respect to even small amounts of noise in the system output vectors. Moreover, the parameter estimates generated by the MAM matrices $\hat{\mathbf{M}}(1)$ were highly inaccurate (discrepancies over two hundred percent) even when σ was set equal to 0 so that the minimum mse value for α was $\alpha^0 = 1.00$. In principle, in accordance with (25) and (26), the MAM matrices $\hat{\mathbf{M}}(\alpha)$ with $\alpha \approx 1.00$ should approximate the memory matrix $\hat{\mathbf{M}} = \mathbf{RS}^{+}$ for the standard linear associative memory procedure outlined in section 3. The present study uses a straightforward Gaussian elimination procedure for matrix inversion which may not be sophisticated enough to calculate $\hat{\mathbf{M}}$ with high accuracy for badly conditioned stimulus matrices $\mathbf{S}$. However, on the basis of other studies (e.g., [8]) in which a more sophisticated MATLAB generalized inverse routine was employed, we expect that the orders of magnitude of the elements of the MAM matrix $\hat{\mathbf{M}}(1)$ accurately reflect the orders of magnitude of the elements of $\hat{\mathbf{M}}$.

Some of these experiments will now be reported in more detail.

Consider, first, the case in which the observation vectors are noise free ($\sigma = 0$), so that the ith observation vector $\mathbf{s}_i^*$ coincides with the ith training output vector $\mathbf{s}_i$. The minimum mse value for α in this case is $\alpha^0 = 1.00$. To what extent are the MAM matrices $\hat{\mathbf{M}}(\alpha)$ able to recover the training parameter vectors $\mathbf{r}_i$ when post-multiplied by the training output vectors $\mathbf{s}_i$?

Table I displays the percentage discrepancies (32) which were obtained at various α-points along the MAM frontier for a Series I experiment with $\sigma = 0$. (The percentage discrepancies appearing in all the tables have been rounded off to the nearest whole number for easier comparison. The training cases i, $i = 1, \ldots, 49$, are numbered from left to right across the rows of each depicted 7×7 training parameter grid, starting with the top row.) For example, in Table I(a) the percentage discrepancies in the estimates for k_0 and θ obtained for the third training case $(k_0, \theta) = (4.0, 0.26)$ at $\alpha = 0.10$ were 2 and 8 percent, respectively, to the nearest whole number. This means that

$$\mathrm{dis}_{3,1}(0.10) = ([\hat{\mathbf{r}}_{3,1}(0.10) - 4.0]/4.0) \times 100 \approx 2\,, \tag{34}$$

Table I. Percentage Discrepancies for the (k_0, θ) MAM Parameter Estimates with Noise-Free Observations.

Table I(a). $\alpha^0 = 1.00$, $\alpha = 0.10$, $C_A = 0.3689$, $C_Z = 0.2999$

	θ						
k_0	0.20	0.23	0.26	0.29	0.32	0.35	0.38
4.0	(1, 32)	(1, 18)	(2, 8)	(2, 0)	(2, −6)	(3, −11)	(4, −15)
4.5	(0, 31)	(0, 18)	(1, 8)	(1, 0)	(2, −6)	(2, −11)	(3, −15)
5.0	(−1, 31)	(0, 17)	(0, 7)	(1, 0)	(1, −6)	(2, −11)	(2, −14)
5.5	(−1, 30)	(−1, 17)	(−1, 7)	(0, 0)	(0, −6)	(1, −10)	(2, −14)
6.0	(−2, 29)	(−1, 16)	(−1, 7)	(−1, −1)	(0, −6)	(1, −10)	(1, −13)
6.5	(−2, 28)	(−2, 16)	(−1, 6)	(−1, −1)	(0, −6)	(0, −10)	(1, −13)
7.0	(−3, 27)	(−2, 15)	(−2, 6)	(−1, −1)	(1, −6)	(0, −10)	(0, −13)

Table I(b). $\alpha^0 = 1.00$, $\alpha = 0.50$, $C_A = 0.0614$, $C_Z = 0.3824$

	θ						
k_0	0.20	0.23	0.26	0.29	0.32	0.35	0.38
4.0	(−1, 30)	(0, 18)	(0, 9)	(0, 3)	(0, −2)	(0, −6)	(1, −9)
4.5	(−1, 27)	(0, 16)	(0, 8)	(0, 2)	(0, −3)	(0, −6)	(1, −8)
5.0	(−1, 24)	(0, 14)	(0, 6)	(0, 1)	(0, −3)	(0, −6)	(1, −8)
5.5	(−1, 21)	(0, 11)	(0, 5)	(0, 0)	(0, −4)	(0, −6)	(1, −8)
6.0	(−1, 17)	(0, 9)	(0, 3)	(0, −1)	(0, −5)	(0, −7)	(1, −8)
6.5	(−1, 14)	(−1, 6)	(0, 1)	(0, −3)	(0, −5)	(0, −7)	(1, −8)
7.0	(−1, 10)	(−1, 4)	(0, −1)	(0, −4)	(0, −6)	(0, −8)	(1, −8)

Table I(c). $\alpha^0 = 1.00$, $\alpha = 0.90$, $C_A = 0.0094$, $C_Z = 0.5151$

	θ						
k_0	0.20	0.23	0.26	0.29	0.32	0.35	0.38
4.0	(0, 18)	(0, 10)	(0, 4)	(0, 0)	(0, −3)	(0, −6)	(0, −7)
4.5	(0, 16)	(0, 9)	(0, 4)	(0, 0)	(0, −2)	(0, −4)	(0, −5)
5.0	(0, 12)	(0, 7)	(0, 3)	(0, 0)	(0, −2)	(0, −3)	(0, −3)
5.5	(0, 9)	(0, 4)	(0, 1)	(0, 0)	(0, −1)	(0, −2)	(0, −1)
6.0	(0, 5)	(0, 2)	(0, 0)	(0, −1)	(0, −1)	(0, −1)	(0, 0)
6.5	(0, 1)	(0, −1)	(0, −2)	(0, −2)	(0, −1)	(0, 0)	(0, 1)
7.0	(0, −3)	(0, −4)	(0, −4)	(0, −3)	(0, −2)	(0, 0)	(0, 2)

Table I(d). $\alpha^0 = 1.00$, $\alpha = 1.00$, $C_A = 0.1270 \times 10^6$, $C_Z = 0.5134 \times 10^{11}$

	θ						
k_0	0.20	0.23	0.26	0.29	0.32	0.35	0.38
4.0	(1103, 635)	(1127, 574)	(1153, 529)	(1181, 496)	(1212, 472)	(1247, 453)	(1286, 440)
4.5	(1002, 580)	(1024, 526)	(1048, 488)	(1075, 460)	(1104, 440)	(1136, 425)	(1172, 415)
5.0	(921, 525)	(942, 479)	(964, 447)	(989, 424)	(1016, 408)	(1047, 397)	(1080, 390)
5.5	(855, 470)	(874, 433)	(895, 406)	(919, 388)	(945, 376)	(973, 369)	(1005, 365)
6.0	(799, 415)	(818, 386)	(838, 366)	(860, 352)	(884, 344)	(912, 340)	(942, 339)
6.5	(752, 360)	(770, 339)	(789, 325)	(810, 316)	(833, 311)	(859, 310)	(888, 313)
7.0	(712, 304)	(729, 292)	(747, 284)	(767, 279)	(789, 278)	(814, 281)	(841, 286)

where the MAM estimate $\hat{\mathbf{r}}_{3,1}(0.10)$ for k_0 is the first component of $\hat{\mathbf{r}}_3(0.10) = \hat{\mathbf{M}}(0.10)\mathbf{s}_3$, and

$$\text{dis}_{3,2}(0.10) = ([\hat{\mathbf{r}}_{3,2}(0.10) - 0.26]/0.26) \times 100 \approx 8\,, \tag{35}$$

where the MAM estimate $\hat{\mathbf{r}}_{3,2}(0.10)$ for θ is the second component of $\hat{\mathbf{r}}_3(0.10) = \hat{\mathbf{M}}(0.10)\mathbf{s}_3$.

Several interesting observations can be made about Table I. First, even when α takes on the value 0.10, and is thus very far from its minimum mse value $\alpha^0 = 1.00$, highly accurate estimates are obtained for the initial capital-labor ratio k_0, especially along the reverse diagonal. Moreover, the corresponding estimates for the capital share θ are highly accurate toward the center of the training grid, and reasonably accurate (less than twenty percent) elsewhere with the exception of the column where θ takes on its smallest training value 0.20. Estimation accuracy improves with increases in α as long as α remains below 1.00. At $\alpha = 1.00$, the components of the MAM matrix take on very large values (note the magnitude of the size cost C_Z), and the resulting parameter estimates are highly inaccurate.

Qualitatively similar results were obtained for Series II experiments with $\sigma = 0$. The discrepancies obtained for the gross savings rate s tended to be higher than those obtained for k_0, whereas the discrepancies obtained for θ tended to be about the same. For comparison purposes, Table II displays the percentage discrepancies (32) which were obtained at $\alpha = 0.90$ for a Series II experiment with $\sigma = 0$.

What happens in the noisy observation case $\sigma > 0$? For the MAM estimates of k_0 in the Series I experiments, the answer is 'not much'. The percentage discrepancies for k_0 for the most part remained well below 10 percent all along the MAM frontier for each σ in the tested range 0.05 to 0.40. Occasionally along the boundary of the training parameter grid the percentage discrepancies rose above 10 percent, but only by a few percentage points. The corresponding MAM estimates for θ were reasonably accurate (discrepancies around twenty percent or

Table II. Percentage Discrepancies for the (s, θ) MAM Parameter Estimates with Noise-Free Observations. $\alpha^0 = 1.00$, $\alpha = 0.90$, $C_A = 0.0213$, $C_Z = 0.2235$

		θ						
		0.20	0.23	0.26	0.29	0.32	0.35	0.38
	0.09	(13, 0)	(11, −4)	(9, −6)	(7, −8)	(5, −9)	(2, −9)	(0, −9)
	0.11	(5, 6)	(4, 2)	(2, 0)	(1, −2)	(0, −3)	(−1, −3)	(−3, −3)
	0.13	(0, 11)	(−1, 6)	(−1, 3)	(−1, 1)	(−2, 0)	(−2, −1)	(−2, −1)
s	0.15	(−3, 14)	(−3, 9)	(−3, 5)	(−2, 3)	(−2, 1)	(−1, 0)	(0, −1)
	0.17	(−5, 16)	(−4, 10)	(−3, 6)	(−2, 3)	(−1, 1)	(1, −1)	(3, −3)
	0.19	(−7, 17)	(−5, 11)	(−3, 6)	(−1, 2)	(1, −2)	(4, −5)	(7, −8)
	0.21	(−7, 17)	(−5, 10)	(−3, 4)	(0, −1)	(3, −6)	(7, −10)	(11, −15)

less) over the interior of the training parameter grid for each tested σ value when the value of α was set roughly in the neighborhood of α^0, e.g., a distance apart of about 0.20 or less.

In the Series II experiments, reasonably accurate estimates (discrepancies around twenty percent or less) were obtained for both s and θ over the interior of the training parameter grid for each tested σ value when the value of α was set roughly near α^0, e.g., a distance apart of about 0.20 or less. Table III displays some of the Series II estimates for α values near α^0 as σ takes on successively higher values.

Table III. Percentage Discrepancies for the (s, θ) MAM Parameter Estimates with Normal $N(0, \sigma^2)$ Observation Noise

Table III(a). $\sigma = 0.10$, $\alpha^0 = 0.67$, $\alpha = 0.70$, $C_A = 0.0527$, $C_Z = 0.0890$

		θ					
	0.20	0.23	0.26	0.29	0.32	0.35	0.38
0.09	(8, 3)	(7, 22)	(12, −2)	(15, −18)	(−2, −5)	(3, −4)	(4, −21)
0.11	(−4, 31)	(6, 2)	(2, 3)	(−2, −3)	(4, −8)	(−1, −18)	(2, −25)
0.13	(−5, 29)	(0, −1)	(−7, −13)	(−5, −7)	(0, −9)	(2, 3)	(−2, −30)
s 0.15	(−6, 25)	(−3, 10)	(0, 4)	(1, 12)	(−2, 12)	(3, 1)	(2, 9)
0.17	(−7, 31)	(−8, 7)	(−1, 11)	(−3, −11)	(0, −7)	(1, −6)	(6, −13)
0.19	(−7, 9)	(−7, 17)	(−7, −1)	(0, 3)	(−1, −15)	(5, −5)	(8, −2)
0.21	(−13, 39)	(−6, 15)	(−4, −6)	(−2, 15)	(1, 1)	(6, −18)	(11, −15)

Table III(b). $\sigma = 0.25$, $\alpha^0 = 0.25$, $\alpha = 0.20$, $C_A = 0.1378$, $C_Z = 0.0045$

		θ					
	0.20	0.23	0.26	0.29	0.32	0.35	0.38
0.09	(12, 10)	(8, 9)	(12, 16)	(2, 4)	(14, −19)	(11, −20)	(8, −29)
0.11	(0, 30)	(−4, 12)	(6, 0)	(3, 2)	(1, −11)	(7, −18)	(7, −25)
0.13	(−2, 49)	(−7, 23)	(−6, 11)	(−9, 7)	(1, −1)	(9, −16)	(1, −25)
s 0.15	(−6, 51)	(−8, 16)	(−7, −3)	(−2, −7)	(0, −5)	(5, −10)	(5, −14)
0.17	(−13, 36)	(−7, 12)	(−6, 6)	(−4, −2)	(−2, −1)	(1, −16)	(10, −16)
0.19	(−11, 39)	(−8, 25)	(−6, 15)	(−1, 5)	(−2, 1)	(13, −8)	(7, −21)
0.21	(−12, 36)	(−10, 26)	(−6, 18)	(−2, 6)	(0, 1)	(4, −7)	(12, −19)

Table III(c). $\sigma = 0.40$, $\alpha^0 = 0.11$, $\alpha = 0.10$, $C_A = 0.1532$, $C_Z = 0.0019$

		θ					
	0.20	0.23	0.26	0.29	0.32	0.35	0.38
0.09	(−4, 35)	(23, 19)	(2, 17)	(−2, −7)	(4, −19)	(12, −22)	(11, −18)
0.11	(3, 42)	(2, 21)	(8, 7)	(10, −5)	(14, −15)	(7, −19)	(14, −27)
0.13	(−6, 35)	(−12, 26)	(−1, 4)	(−10, −9)	(4, −7)	(−5, −24)	(11, −24)
s 0.15	(−3, 43)	(−10, 34)	(−2, 7)	(−3, 6)	(1, −13)	(8, −11)	(−1, −23)
0.17	(−14, 48)	(−12, 32)	(−8, 17)	(4, −4)	(7, −3)	(0, −14)	(5, −15)
0.19	(−13, 23)	(−9, 35)	(−14, 11)	(3, −10)	(1, −2)	(2, −22)	(13, −16)
0.21	(−10, 47)	(−13, 28)	(−6, 13)	(−1, 1)	(2, −8)	(3, −9)	(2, −22)

6.3. EXPERIMENTAL RESULTS WITH CORRELATED OBSERVATION NOISE

The findings reported in the previous subsection suggest that the MAM matrices $\hat{\mathbf{M}}(\alpha)$ are successfully able to reject i.i.d. mean-zero noise in the observation vectors if α is set equal to a value in [0.10, 0.99] which is even roughly near the minimum mse value α^0 in (33). What happens when the observation noise is correlated? In particular, will the simple MAM procedure still deliver usable parameter estimates along at least some portions of the MAM frontier, or will a more sophisticated memory matrix construction be required?

To gain some preliminary understanding of this issue, the 15×1 training output vectors $\mathbf{s}_i$, $i = 1, \ldots, 49$, were corrupted with correlated sinusoidal noise. Specifi-

Table IV. Percentage Discrepancies for the (s, θ) MAM Parameter Estimates with Sinusoidal Observation Noise of Amplitude A

Table IV(a). $A = 0.10$, $\alpha = 0.70$, $C_A = 0.0527$, $C_Z = 0.0890$

	θ						
s	0.20	0.23	0.26	0.29	0.32	0.35	0.38
0.09	(11, 9)	(10, 2)	(7, 0)	(8, −9)	(5, −9)	(8, −14)	(6, −19)
0.11	(3, 17)	(3, 6)	(1, 4)	(0, 1)	(−7, 13)	(−7, 10)	(0, −11)
0.13	(−2, 20)	(−1, 11)	(−3, 9)	(−1, −4)	(−5, −3)	(1, −9)	(1, −10)
0.15	(−5, 26)	(−4, 15)	(−2, 1)	(2, 4)	(−2, 3)	(1, −8)	(2, −9)
0.17	(−6, 24)	(−6, 19)	(−2, 2)	(1, −11)	(−1, 0)	(3, −8)	(5, −9)
0.19	(−8, 28)	(−7, 29)	(−4, 20)	(−1, 0)	(1, −5)	(5, −9)	(9, −14)
0.21	(−8, 28)	(−4, 18)	(−5, 13)	(0, 0)	(2, −4)	(7, −13)	(12, −15)

Table IV(b). $A = 0.25$, $\alpha = 0.20$, $C_A = 0.1378$, $C_Z = 0.0045$

	θ						
s	0.20	0.23	0.26	0.29	0.32	0.35	0.38
0.09	(8, 29)	(8, 14)	(6, 4)	(8, −7)	(8, −13)	(18, −22)	(12, −26)
0.11	(0, 33)	(1, 17)	(0, 7)	(1, −2)	(−2, −4)	(−1, −6)	(5, −23)
0.13	(−4, 37)	(−3, 20)	(−3, 9)	(−1, −2)	(−9, −9)	(4, −17)	(5, −22)
0.15	(−7, 39)	(−5, 22)	(−2, 8)	(8, 4)	(0, −6)	(3, −15)	(5., −19)
0.17	(−8, 40)	(−7, 25)	(−2, 4)	(0, −2)	(0, −6)	(4, −13)	(7, −18)
0.19	(−9, 24)	(−8, 31)	(−2, 18)	(−2, 3)	(1, −5)	(5, −12)	(9, −18)
0.21	(−10, 44)	(−2, 33)	(−6, 17)	(−1, 5)	(2, −3)	(7, −12)	(12, −17)

Table IV(c). $A = 0.40$, $\alpha = 0.10$, $C_A = 0.1532$, $C_Z = 0.0019$

	θ						
s	0.20	0.23	0.26	0.29	0.32	0.35	0.38
0.09	(9, 31)	(9, 16)	(7, 4)	(10, −6)	(9, −25)	(24, −25)	(14, −26)
0.11	(1, 34)	(2, 19)	(1, 7)	(1, −4)	(−1, −9)	(0, −9)	(7, −25)
0.13	(−5, 39)	(−3, 22)	(−3, 9)	(−2, 0)	(−14, −11)	(5, −18)	(6, −23)
0.15	(−8, 41)	(−5, 22)	(−2, 10)	(13, 5)	(0, −8)	(4, −5)	(5, −20)
0.17	(−8, 43)	(−7, 25)	(−1, 2)	(0, 2)	(0, −7)	(4, −14)	(7, −19)
0.19	(−10, 44)	(−7, 29)	(0, 16)	(−2, 4)	(1, −5)	(5, −12)	(9, −18)
0.21	(−11, 46)	(1, 38)	(−6, 16)	(−2, 6)	(1, −2)	(6, −10)	(11, −18)

cally, for each training case i we constructed a 15×1 noise vector $\mathbf{n}_i$ with jth component given by

$$n_{ij} = A \sin(ij + i^2) . \tag{36}$$

Given any positive value for the amplitude A, the noise components (36) are highly correlated. For example, with an amplitude $A = 0.05$, the components of the noise vector $\mathbf{n}_1$ cycle through positive and negative values as follows:

$$+0.0455, +0.0071, -0.0378, -0.0479, -0.0140, +0.0328, +0.0495, +0.0206,$$
$$-0.0272, -0.0500, -0.0268, +0.0211, +0.0495, +0.0325, -0.0144 . \tag{37}$$

The training output vectors $\mathbf{s}_i$ were then corrupted with these noise terms to form noisy observation vectors $\mathbf{s}_i^* = [\mathbf{s}_i + \mathbf{n}_i]$. As in Subsection 6.2, tests were conducted to determine the extent to which each MAM matrix $\hat{\mathbf{M}}(\alpha)$ constructed using the noise-free training vectors $\{(\mathbf{r}_1, \mathbf{s}_1), \ldots, (\mathbf{r}_{49}, \mathbf{s}_{49})\}$ was successfully able to recover the training parameter vector $\mathbf{r}_i$ when post-multiplied by the noisy observation vector $\mathbf{s}_i^*$, $i = 1, \ldots, 49$. Recoverability was investigated for amplitudes A ranging from 0 to 0.40, where, in each experiment, $A = 0.zx$ represented a measurement error of about $zx/2$ percent (cf. Subsection 6.2).

Surprisingly, for both the Series I and Series II experiments, the MAM matrices $\hat{\mathbf{M}}(\alpha)$ were able to recover the training parameter vectors from the observation vectors corrupted with correlated noise with the same degree of accuracy they displayed in the Series I and II experiments with i.i.d. noise. Comparing experiments with the same σ and A values (i.e., experiments with approximately the same size measurement errors), the magnitudes of the discrepancies obtained over the training grids for each tested α value are remarkably similar, column by column.

Table IV displays some of the Series II estimates for α values near α^0 as the amplitude A takes on successively higher values. The results in Table III(a) can be directly compared with the results in Table IV(a),* and similarly for parts (b) and (c) in these tables.

7. Concluding Remarks

The successful implementation of successive approximation procedures for nonlinear least squares problems typically requires initial estimates for the parameter vector which are within ten or twenty percent of the actual solution vector. The

* The training parameter grid is the same in Tables III(a) and IV(a), hence the same MAM matrix $\hat{\mathbf{M}}(0.10)$ is constructed based on the same set of noise-free training vectors. The association and size costs C_A and C_Z are therefore also the same. The only differene between the two tables is that two different kinds of noise (i.i.d. normal noise versus correlated sinusoidal noise) are used to corrupt the observation vectors prior to obtaining the MAM parameter estimates. Consequently, the differences in the displayed discrepancy terms are entirely due to the different types of noise.

experimental results reported in this paper suggest that one or more usable initial estimates for the parameter vector might be found by considering the MAM parameter vector estimates corresponding to a rough sample of α-points along the MAM frontier.

Specifically, in each experiment with observation noise, whether correlated or i.i.d., reasonably accurate MAM parameter estimates were obtained over the interior of the training parameter grid for α-values in $[0.10, 0.99]$ lying even roughly nearby the (generally unknown) 'optimal' alpha value (33). That is, the discrepancies between these estimates and the true parameter values were around twenty percent or less. On the other hand, even in the absence of observation noise, highly inaccurate parameter estimates were obtained when α was set equal to 1.00, i.e., when the standard linear associative memory procedure was used.

In addition, for each tested α value in $[0.10, 0.99]$, the MAM parameter estimates lying near the center of the training parameter grid tended to be highly accurate. The latter finding poses a fixed point problem for the data analyst: Find a training parameter grid so that the resulting MAM matrix constructed on the basis of this grid maps the observation vector back into the center of the grid. A constructive way to solve this fixed point problem might be to turn the MAM procedure into an iterative procedure by successively recentering the training parameter grid around the last MAM estimate one has obtained.

Much additional theoretical and simulation work is, of course, needed to understand and to expand upon these experimental findings. Nevertheless, the results obtained to date suggest that multicriteria associative memories provide a powerful new tool for nonlinear estimation.

Acknowledgements

Partially supported by NIH Grant No. DK 33729. A preliminary version of this paper was presented at the 1990 Meeting of the Society for Economic Dynamics and Control, University of Minnesota, St. Paul Campus, 29 June, 1990, and at the Neural Network Seminar, Computer Science Department, ISU, 5 December, 1990. The authors are grateful to K. Judd, J.-F. Richard, and an anonymous referee for helpful comments, and to H.C. Quirmbach for computer graphics. Please address correspondence to L. Tesfatsion.

References

1. Albert, A., Regression and the Moore–Penrose Pseudoinverse, Academic Press, New York, 1972.
2. Allgower, E. and Georg, K., Simplicial and continuation methods for approximating fixed points and solutions to systems of equations, *SIAM Rev.* 22 (1980), 28–85.
3. Bates, D.M. and Watts, D.G., *Nonlinear Regression Analysis and its Applications*, Wiley, New York, 1988.
4. Cuyt, A. and Rall, L., Computational implementation of the multivariate Halley method for solving nonlinear systems of equations, *ACM Trans. Math. Software* **11** (1985), 20–36.

5. Denison, E., *Trends in American Economic Growth*, 1929–1982, The Brookings Institution, Washington, D.C., 1985.
6. Judge, G., Griffiths, W., Hill, R., Lutkepohl, H. and Lee, T.-C., *The Theory and Practice of Econometrics*, Wiley, New York, 1985.
7. Kalaba, R. and Tesfatsion, L., Solving nonlinear equations by adaptive homotopy continuation, *Appl. Math. Comput.* **41** (1991), 99–115.
8. Kalaba, R., Lichtenstein, Z., Simchony, T. and Tesfatsion, L., Linear and nonlinear associative memories for parameter estimation, MRG Working Paper 8913, Department of Economics, USC, Los Angeles, September 1989, to appear in *Inform. Sci.*
9. Kalaba, R., Kim, M. and Moore II, J., Linear programming and simple associative memories, *Appl. Math. Comput.* **40** (1990), 203–214.
10. Kohonen, T., *Self-Organization and Associative Memory*, Springer-Verlag, New York, 1988.
11. Kosko, B., Bidirectional associative memories, *IEEE Trans. Systems Man Cybernet.* **SMC-18** (1988), 49–60.
12. Miller, J., Parameter identification by continuation methods, *Appl. Math. Comput.* **34** (1989), 17–27.
13. Murakami, K. and Aibara, T., An improvement on the Moore-Penrose generalized inverse associative memory, *IEEE Trans. Systems Man Cybernet.* **SMC-17** (1987), 699–706.
14. Olivier, P., Optimal noise rejection in linear associative memories, *IEEE Trans. Systems Man Cybernet.* **SMC-18** (1988), 814–815.
15. Penrose, R., On best approximate solutions of linear matrix equations, *Proc. Cambridge Phil. Soc.* **52** (1956), 17–20.
16. Poggio, T., On optimal nonlinear associative recall, *Biol. Cybernet.* **19** (1975), 201–209.
17. Prescott, E., Robert M. Solow's neoclassical growth model: an influential contribution to economics, *Scand. J. Econom.* **90** (1988), 7–12.
18. Ramanathan, R., *Introduction to the Theory of Economic Growth*, Springer-Verlag, New York, 1982.
19. Rumelhart, D. and McClelland, J. (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vols. 1 and 2, MIT Press, Cambridge, Mass. 1986.
20. Scales, L., *Introduction to Non-Linear Optimization*, Springer-Verlag, New York, 1985.
21. Simon, J., Great and almost-great magnitudes in economics, *J. Econom. Perspectives* **4** (1990), 149–156.
22. Vinod, H. and Ullah, A., *Recent Advances in Regression Methods*, Marcel Dekker, New York, 1981.
23. Watson, L., Billups, T. and Morgan, A., G52: HOMPACK: A suite of codes for globally convergent homotopy algorithms, *ACM Trans. Math. Software* **13** (1987), 281–310.