ON THE OPTIMAL CONSISTENT APPROXIMATION TO PAIRWISE COMPARISON MATRICES

MOODY T. CHU*

Abstract. Consistency retrieval from a biased relative preference table is an imperative task in decision theory. This paper considers the least squares approximation of a pairwise comparison matrix by consistent matrices. It is observed that the highly nonlinear manifold of consistent matrices can be changed into a linear subspace by the component-wise logarithmic transformation. A first order optimality condition therefore can be described in terms of coordinates in the linear subspace. This approach facilitates the otherwise much more complicated optimality condition if working with the variables in the original manifold. Fast nonlinear equation solvers can be employed to solve the problem efficiently.

Key words. Decision Making, Priority Setting, Pairwise Comparisons, Consistency, Least Squares, Quasi-Newton Algorithm.

AMS(MOS) subject classifications. 65F15, 65H15.

1. Introduction.

Decision making, especially on intangible stimuli or criteria such as the degree of environmental hazard factors or psychological impact factors, is a very hard task. Not only the information about the stimuli is often inexact or incomplete, but also the decision maker's own judgment is sometimes inconsistent. Given n stimuli, one way to acquire better insights into the underlying system is to assign a *weight* of priority to each stimulus and to compare the stimuli in pairs. In the ideal situation where there are exact positive values $v_1, \ldots, v_n$ for the stimuli, the quotient $m_{ij} = v_i/v_j$, called the *relative preference* of stimulus i to stimulus j in the literature, can be used as a powerful inference tool in knowledge-based or data mining expert systems. Practical and theoretical discussion of the knowledge acquisition process based on the method of pairwise comparison can be found in [4, 7].

In practice, however, it is difficult to establish the relative preference matrix $[m_{ij}]$ exactly either because a priority setting $v_1, \ldots, v_n$ cannot be possibly measured or because such a weight estimation itself is inexact. In the decision making procedure quite often the pairwise comparison coefficients m_{ij} are provided through some other avenues and are meant only to be an approximation to the true yet unknown quotients v_i/v_j. Under circumstances such as this, one important issue stands out before a relative preference table can be used to help decision making, i.e., the inconsistency embedded in the estimated pairwise comparison coefficients must be removed or reduced.

To describe the problem more precisely, we say that $M = [m_{ij}] \in R^{\times n}$ is a *pairwise comparison* matrix if $m_{ij} > 0$ for all $i, j = 1, \ldots, n$. (So a pairwise comparison matrix is in fact a positive matrix as we normally call in the literature.) A pairwise comparison matrix M is called *consistent* if

$$m_{ik}m_{kj} = m_{ij} \tag{1}$$

* Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205. This research was supported in part by the National Science Foundation under grant DMS-9123448.

for all $i, j, k = 1, \ldots, n$. Note that a consistent matrix M is necessarily *reciprocal*, i.e., $m_{ji} = \frac{1}{m_{ij}}$ for all i, j, but the converse is not true in general. Consistent matrices correspond to the ideal situation in which there are exact values $v_1, \ldots, v_n$ for the stimuli. The quotients $m_{ij} = v_i/v_j$ then form a consistent matrix.

The pairwise comparison matrix arising in practice, due to noise or imperfect judgments, usually is not consistent. The challenge is to best approximate a given pairwise comparison matrix Z by a consistent matrix in some sensible way. Several approaches have been proposed. Motivated by the Frobenius Theorem, for example, Saaty [6] suggests that the eigenvector $v = [v_1, \ldots, v_n]^T$ corresponding to the largest eigenvalue in modulus of Z would be a reasonable priority setting. Crawford, on the other hand, proposes a geometric means procedure for estimating the scale of judgment matrix [2]. A Monte Carlo study comparing the performance of these two methods can be found in [4]. This paper discusses the best consistent approximation of Z in the sense of least squares. We outline a procedure using the quasi-Newton method to solve the problem.

It would be wrong to simply consider the problem

$$\min_{m_{ji}=\frac{1}{m_{ij}}} \|Z - [m_{ij}]\|_F$$

because the solution matrix, though is reciprocal, may not be consistent. It would also be impractical because there are $\frac{n(n-1)}{2}$ variables involved in the above optimization whereas the relationship (1) implies, by induction, that a consistent matrix $M = [m_{ij}]$ must be of the form

$$(2) \qquad m_{ij} = \begin{cases} 1, & \text{if } j = i, \\ s_i s_{i+1} \cdots s_{j-1}, & \text{if } j > i, \\ \frac{1}{s_i s_{i+1} \cdots s_{j-1}}, & \text{if } j < i, \end{cases}$$

where $s_1, \ldots, s_{n-1}$ are some positive numbers. Clearly, the consistent matrices form a much more complicated nonlinear submanifold of dimension $n-1$.

The relationship in (2) defines one way to parameterize the manifold of consistent matrices, i.e., by using values $s_1, \ldots, s_{n-1}$ of the sup-diagonal entries of the matrix. By working with these parameters, one may therefore formulate the best approximation as the solution to the least squares problem

$$(3) \qquad \min_{s_1>0,\ldots,s_{n-1}>0} \|Z - M(s_1, \ldots, s_{n-1})\|_F$$

where entries of $M = M(s_1, \ldots, s_{n-1})$ are defined by (2). It can be shown that the resulting optimality condition, though complicated, is equivalent to yet another simpler way of parameterization discussed in this paper. Our main point is that the manifold of consistent matrices can be nicely parameterized through the following logarithmic transformation that facilitates the derivation of the optimality condition and the computation of the projected gradient.

Consider the set

$$(4) \qquad \mathcal{L} := \left\{ L = [\ell_{ij}] \in R^{n \times n} | \ell_{ik} + \ell_{kj} = \ell_{ij} \text{ for all } i, j, k \right\}.$$

Clearly $\mathcal{L}$ forms a linear subspace. It is important to notice that corresponding to each $L \in \mathcal{L}$ the matrix

$$\exp(L) := [e^{\ell_{ij}}], \tag{5}$$

i.e., the element-wise exponential of L, is a consistent matrix. Similarly, corresponding to each consistent matrix M the matrix

$$\log(M) := [\ln m_{ij}], \tag{6}$$

i.e., the element-wise logarithm of M, is an element in $\mathcal{L}$. Furthermore, the corresponding is a one-to-one mapping. Given a pairwise comparison matrix Z, the least squares approximation problem can now be stated as:

$$\min_{L \in \mathcal{L}} \|Z - \exp(L)\|_F. \tag{7}$$

2. Optimality Condition. Suppose $L \in \mathcal{L}$. It is clear that $\ell_{ii} = 0$ for all i. The following theorem states that L is completely characterized by its last column.

THEOREM 2.1. *Let the last column of L be denoted by $[t_1, \ldots, t_{n-1}, 0]^T$. Then $L = [\ell_{ij}]$ is determined by the rule*

$$\ell_{ij} = t_i - t_j. \tag{8}$$

Proof. Since $\ell_{nj} + \ell_{jn} = \ell_{nn} = 0$, it follows that $\ell_{nj} = -t_j$ for all j. Then $\ell_{ij} = \ell_{in} + \ell_{nj} = t_i - t_j$. □

We shall denote L by $L(t_1, \ldots, t_{n-1})$. It follows immediately that $\mathcal{L}$ is of dimension $n-1$. A natural basis for $\mathcal{L}$ would be $\{L^{(1)}, \ldots L^{(n-1)}\}$ where

$$L^{(i)} := L(e_k) \tag{9}$$

and e_k is the standard unit vector $[0, \ldots, 1, \ldots 0]^T$ in R^{n-1} with 1 at its k-th position. It is easy to see that the entries $\ell_{ij}^{(k)}$ of $L^{(k)}$ are given by

$$\ell_{ij}^{(k)} = \begin{cases} 1, & \text{if } i = k \text{ and } j \neq k; \\ -1, & \text{if } j = k \text{ and } i \neq k; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

As a linear subspace, $\mathcal{L}$ may be characterized by many other ways than using the last column of a matrix. For example, (2) suggests that we may use the sup-diagonal of $L \in \mathcal{L}$ to delineate the matrix, i.e., we may write $L = [\ell_{ij}]$ as

$$\ell_{ij} = \begin{cases} 0, & \text{if } j = i, \\ s_i + s_{i+1} + \cdots + s_{j-1}, & \text{if } j > i, \\ -s_i - s_{i+1} - \cdots - s_{j-1}, & \text{if } j < i, \end{cases}$$

where $\{s_1, \ldots, s_{n-1}\}$ are sup-diagonal entries of L. The resulting basis however are unnecessarily more complicated. We think that the basis defined by (10) is perhaps the simplest for $\mathcal{L}$ and, as will be seen below, simplifies the calculation.

The element-wise exponential function exp defined in (5) is differentiable. Note that

$$\begin{aligned} \exp(M+H) - \exp(M) &= [e^{m_{ij}+h_{ij}} - e^{m_{ij}}] \\ &= [e^{m_{ij}} h_{ij}] + O([h_{ij}^2]). \end{aligned}$$

It follows that the Fréchet derivative of exp at M acting on H is given by

$$\exp'(M)(H) = \exp(M) \circ H \tag{11}$$

where $A \circ B = [a_{ij} b_{ij}]$ stands for the Hadamard product of matrices A and B. Define

$$F(L) := \frac{1}{2} \langle Z - \exp(L), Z - \exp(L) \rangle \tag{12}$$

where $\langle A, B \rangle = \sum_{i,j} a_{ij} b_{ij}$ stands for the Frobenius inner product of matrices A and B. We can calculate the gradient of F according to the following theorem.

THEOREM 2.2. *With respect to the Frobenius inner product, the gradient ∇F is given by*

$$\nabla F(L) = -(Z - \exp(L)) \circ \exp(L). \tag{13}$$

Proof. By (11), we find that

$$\begin{aligned} F'(L)(H) &= \langle Z - \exp(L), -\exp(L) \circ H \rangle \\ &= \langle -(Z - \exp(L)) \circ \exp(L), H \rangle. \end{aligned} \tag{14}$$

In the second equation above, we have used the fact that

$$\langle A, B \circ H \rangle = \sum_{i,j} a_{ij}(b_{ij} h_{ij}) = \sum_{ij} (a_{ij} b_{ij}) h_{ij} = \langle A \circ B, H \rangle.$$

The assertion therefore follows from (14). □

We can easily project $\mathrm{Proj}_{\mathcal{L}} \nabla F(L)$ of the gradient $\nabla F(L)$ onto the space $\mathcal{L}$. The projection is necessarily a linear combination of the basis $L^{(k)}$ defined in (9). Suppose

$$\mathrm{Proj}_{\mathcal{L}} \nabla F(L) = \sum_{k=1}^{n-1} \alpha_k L^{(k)}. \tag{15}$$

Then it follows that

$$\langle \nabla F(L) - \sum_{k=1}^{n-1} \alpha_k L^{(k)}, L^{(i)} \rangle = 0, \quad \text{for } i = 1, \ldots, n-1. \tag{16}$$

That is, $\alpha := [\alpha_1, \ldots, \alpha_{n-1}]^T$ must satisfy the linear system of equations

$$\sum_{k=1}^{n-1} \langle L^{(i)}, L^{(k)} \rangle \alpha_k = \langle \nabla F(L), L^{(i)} \rangle, \quad \text{for } i = 1, \ldots, n-1. \tag{17}$$

The coefficient matrix $\Omega := [\langle L^{(i)}, L^{(k)} \rangle]$ in (17) has very simple form

$$\Omega = \begin{bmatrix} 2(n-1) & -2 & \ldots & -2 & -2 \\ -2 & 2(n-1) & \ldots & & -2 \\ \vdots & & \ddots & & \vdots \\ -2 & & & 2(n-1) & -2 \\ -2 & -2 & & -2 & 2(n-1) \end{bmatrix}.$$

Since $L^{(1)}, \ldots L^{(n-1)}$ are linearly independent, Ω must be nonsingular. In fact, the inverse has very simple closed form,

$$\Omega^{-1} = \frac{1}{n} \begin{bmatrix} 1 & \frac{1}{2} & \ldots & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & & & \\ \vdots & & \ddots & & \\ \frac{1}{2} & & & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \ldots & \frac{1}{2} & 1 \end{bmatrix}. \tag{18}$$

So the computation of α is easy.

Once α is determined, the vector $-\text{Proj}_{\mathcal{L}} \nabla F(L)$ offers a steepest descent search direction in the space $\mathcal{L}$ to decrease the values of F. Together with a line search strategy, we have in hands a descent method that works directly in terms of the variable L.

On the other hand, our approach provides an easy derivation of the first order optimality condition in terms of the parameters $t_1, \ldots, t_{n-1}$.

THEOREM 2.3. *Assume $t_n = 0$. For $L = L(t_1, \ldots, t_{n-1}) \in \mathcal{L}$ to be a local minimizer of F, the system of nonlinear equations*

$$\sum_{j=1}^{n} \left[(z_{kj} - e^{t_k - t_j}) e^{t_k - t_j} - (z_{jk} - e^{t_j - t_k}) e^{t_j - t_k} \right] = 0, \; k = 1, \ldots, n-1 \tag{19}$$

must be satisfied.

Proof. It follows from (15) and (17) that the first order optimality condition for L is

$$\langle (Z - \exp(L)) \circ \exp(L), L^{(k)} \rangle = 0, \quad \text{for } k = 1, \ldots, n-1.$$

The structure of $L^{(k)}$ (see (10)) implies that $\langle A \circ B, L^{(k)} \rangle = \sum_j (a_{kj} b_{kj} - a_{jk} b_{jk})$ for any matrices A and B. The assertion follows after noting that $L = [e^{t_i - t_j}]$. □

These conditions are mathematically equivalent to but operationally simpler than those obtained, for instance, from (2). We illustrate our point by one simple example where $n = 3$ and

$$Z = \begin{bmatrix} 1 & 2 & 5 \\ 4 & 1 & 3 \\ 9 & 1 & 1 \end{bmatrix}. \tag{20}$$

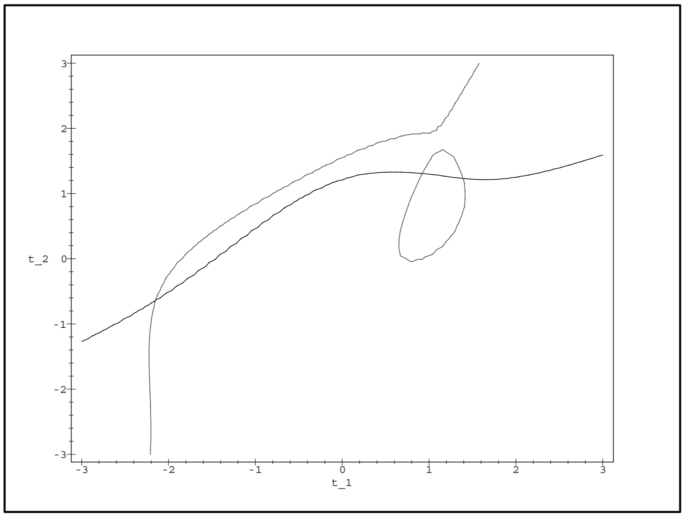

FIG. 1. *Solution curves of each individual equation in (19) with Z given by (20).*

Figure 1 depicts the solution curves (t_1, t_2) to each of the two equations defined in (19). It is seen that there are three simultaneous solutions to the system (19). On the other hand, we can also use the parameterization (2) to solve the problem. Upon differentiating the objective function in (3) with respect to the parameters s_1 and s_2, we obtain a new set of gradient equations whose solution curves are plotted in Figure 2. It is seen that there are six simultaneous solutions to this gradient system. It is conceivable that the higher the dimension n is, the more extraneous solutions there will be. Indeed, the resulting nonlinear system of gradient equations by using the parameterization (2) will in general end up with a system of *Laurent polynomials.* There is a well known theory, Bernstein's Theorem, that predicts the number of solutions by using the mixed volume of the Newton polytopes of the equations. More details can be found in [1, §7.5]. Of course, the problem (3) requires both $s_1 > 0$ and $s_2 > 0$ and, hence, limits the solutions of the above example to the three in the first quadrant that, in return, give the same answers as would be by solving (19). But the fact that our parametrization avoids the calculation of the extraneous points from the beginning is quite remarkable.

3. Numerical Experiment. The unknowns $t_1, \ldots, t_{n-1}$ in (19) can be solved by any available nonlinear equation solver. The resulting matrix $\exp(L(t_1, \ldots, t_{n-1})$ would be regarded as a least squares solution to problem (7) with impunity. In particular, through our parameterization the optimization (7) now becomes an unconstrained

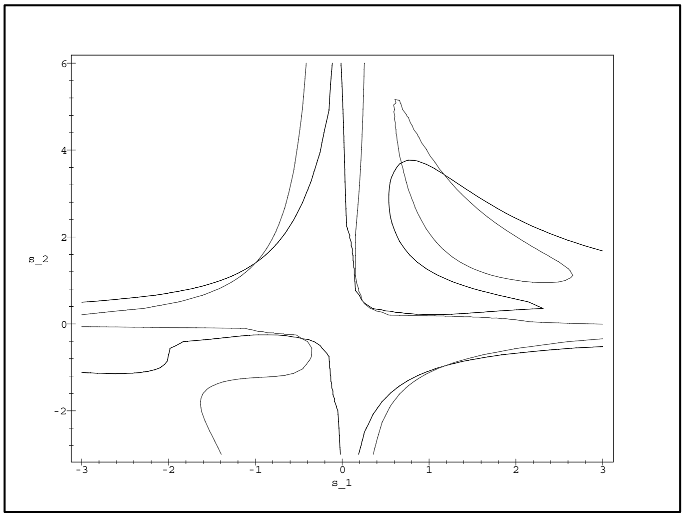

FIG. 2. *Solution curves of each gradient equation obtained from (3) with Z given by (20).*

problem

$$\min_{t_1,\dots,t_{n-1}} G(t_1,\dots,t_{n-1}) := \frac{1}{2}\|Z - \exp(L(t_1,\dots,t_{n-1})\|_F^2. \tag{21}$$

We make a crucial observation that the nonlinear system i in Theorem 2.3 corresponds exactly to the gradient of G.

THEOREM 3.1. *For $k = 1,\dots,n-1$,*

$$\frac{\partial G}{\partial t_k} = \langle -\left(Z - \exp(L(t_1,\dots,t_{n-1}))\right) \circ \exp(L(t_1,\dots,t_{n-1})), L^{(k)}\rangle. \tag{22}$$

Proof. The proof follows directly from the chain rule that $\frac{\partial G}{\partial t_k} = F'(L)\frac{\partial L}{\partial t_k}$, the relationship (14), and the fact that $\frac{\partial L}{\partial t_k} = L^{(k)}$. □

The gradient information therefore can be used to build up curvature information through, for example, the BFGS technique. In return, a search direction can be determined to decrease the value of G. Followed by an appropriate line search procedure, a quasi-Newton algorithm can be developed to solve the least squares problem effectively. There are many readily available library routines for doing this job. We find that the routine **fminu** in MATLAB's Optimization Toolbox [3] is particularly convenient because the MATLAB command **exp** does precisely the component-wise exponential defined in (5). The gradient in (22), for example, can quickly be calculated through the following program:

```
function g=grad(T);
global Z
n=length(T)+1;t=[T;0];
L=diag(t)*ones(n)-ones(n)*diag(t);
temp=-(Z-exp(L)).*exp(L);
temp=sum(temp'-temp);
g=temp(1:n-1);
end
```

We report some numerical experiments in this section. For convenience, we display all numbers only with five digits although all tests are run with a much higher termination criteria for the worst case precision of both the independent variables and the objective function, i.e., the **options** vector in **fminu** is reset so that **options(2)** $= 10^{-8}$ and **options(3)** $= 10^{-8}$. For practical applications, this precision is far better than needed. Also, we have tested some larger size matrices (with n up to 100) and the algorithm performs reasonably well. We concentrate on the case $n = 7$ in this report to illustrate our point.

We note first that the least squares consistent approximation to a given Z is not unique due to the nonlinearity of the problem. Likewise, a solution to the system (19) only satisfies a necessary condition. We have already illustrated that when Z is given by (20) there are three three simultaneous solutions to the system. But upon checking, only the two pairs $(-2.1504, -0.6319)$ and $(1.3964, 1, 2347)$ for (t_1, t_2) correspond to a least squares solution. The third pair $(1.3109, 0.9220)$ solves the nonlinear system (19), but is not a least squares solution. The starting point determines which least squares solution the algorithm converges to.

Example 1. To illustrate the sensitivity of the consistent matrices subject to perturbations, consider the case where a system of seven stimuli has been given priorities $s_i = i$ for $i = 1, \ldots, 7$. The ideal pairwise comparison matrix should be $S = [s_i/s_j]$. Suppose now random noises from a normal distribution of mean 0 and variance 10^{-4} have been added to the off-diagonal entries (common sense would tell that something is obviously wrong if the diagonal entries are not 1's) of S to produce

$$Z = \begin{bmatrix} 1.0000 & 0.4898 & 0.3261 & 0.2613 & 0.2003 & 0.1635 & 0.1180 \\ 1.9890 & 1.0000 & 0.6622 & 0.5015 & 0.3819 & 0.3320 & 0.2973 \\ 3.0112 & 1.5152 & 1.0000 & 0.7570 & 0.6103 & 0.5062 & 0.4183 \\ 4.0058 & 2.0075 & 1.3312 & 1.0000 & 0.8039 & 0.6764 & 0.5830 \\ 4.9973 & 2.4949 & 1.6723 & 1.2702 & 1.0000 & 0.8222 & 0.7064 \\ 6.0041 & 3.0089 & 1.9894 & 1.5092 & 1.2087 & 1.0000 & 0.8635 \\ 6.9902 & 3.4975 & 2.3368 & 1.7319 & 1.4175 & 1.1671 & 1.0000 \end{bmatrix}.$$

Now it is not clear how valid the pairwise comparison matrix Z represents the true relative preference. Starting with the random vector

$$[0.2190, 0.0470, 0.6789, 0.6793, 0.9347, 0.3835]$$

as initial values for $[t_1, \ldots t_6]$, the routine **fminu** takes 27 steps to reduce the gradient to less than 10^{-8}. The history of the objective values are plotted in Figure 3. We make

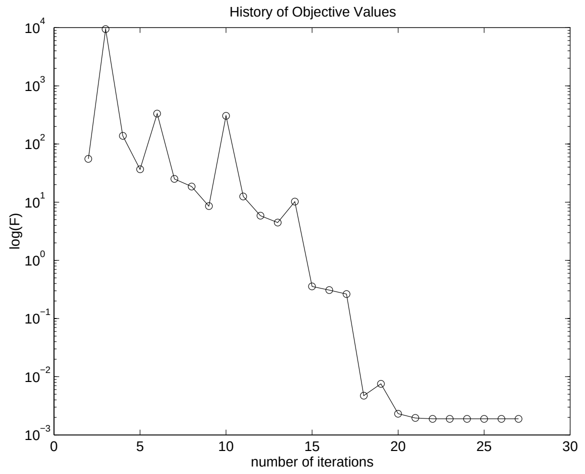

FIG. 3. *History of objective values for Example 1.*

a remark here that this example also represents a typical run of **fminu** on many other experiments we have conducted. It is interesting to note that the best consistent matrix approximation to Z is given by

$$C = \begin{bmatrix} 1.0000 & 0.5009 & 0.3325 & 0.2498 & 0.2002 & 0.1665 & 0.1430 \\ 1.9966 & 1.0000 & 0.6638 & 0.4986 & 0.3997 & 0.3324 & 0.2855 \\ 3.0078 & 1.5065 & 1.0000 & 0.7512 & 0.6022 & 0.5008 & 0.4302 \\ 4.0040 & 2.0054 & 1.3312 & 1.0000 & 0.8017 & 0.6667 & 0.5726 \\ 4.9946 & 2.5016 & 1.6605 & 1.2474 & 1.0000 & 0.8316 & 0.7143 \\ 6.0057 & 3.0080 & 1.9967 & 1.4999 & 1.2024 & 1.0000 & 0.8589 \\ 6.9924 & 3.5022 & 2.3247 & 1.7464 & 1.4000 & 1.1643 & 1.0000 \end{bmatrix}.$$

It is important to note that C is closer to S than to Z, i.e, $\|C - S\|_F = 0.0233$ and $\|C - Z\|_F = 0.0615$ whereas $\|Z - S\|_F = 0.0658$, indicating that C is really retrieving consistency from the perturbed S.

Example 2. We repeat the above experiment 400 times with consistent matrices S whose weights of priority are generated randomly by the absolute value of **randn**, a MATLAB random number generator of uniform distribution with mean 0 and variance 1. Random noises from **randn*0.01** are added to produce testing data Z. Initial guesses for $t_1, \ldots t_6$ are also randomly generated from **randn**. We plot in Figure 4 the differences among S, Z, and C measured by the Frobenius norm. It is observed that $\|C - S\|_F$ is generally two to three times smaller than $\|C - Z\|_F$.

4. Conclusion.

In the process of knowledge acquisition, one important approach is to introduce weights reflecting the relative significance of the objectives concerned. In reality, however, these weights either cannot be precisely assigned or are assigned with biased judgments. We have discussed in this paper an important issue of retrieving consistency from the data that are in disarray. We propose a special parameterization that in conjunction with the quasi-Newton method enables us to carry out this validation process effectively.

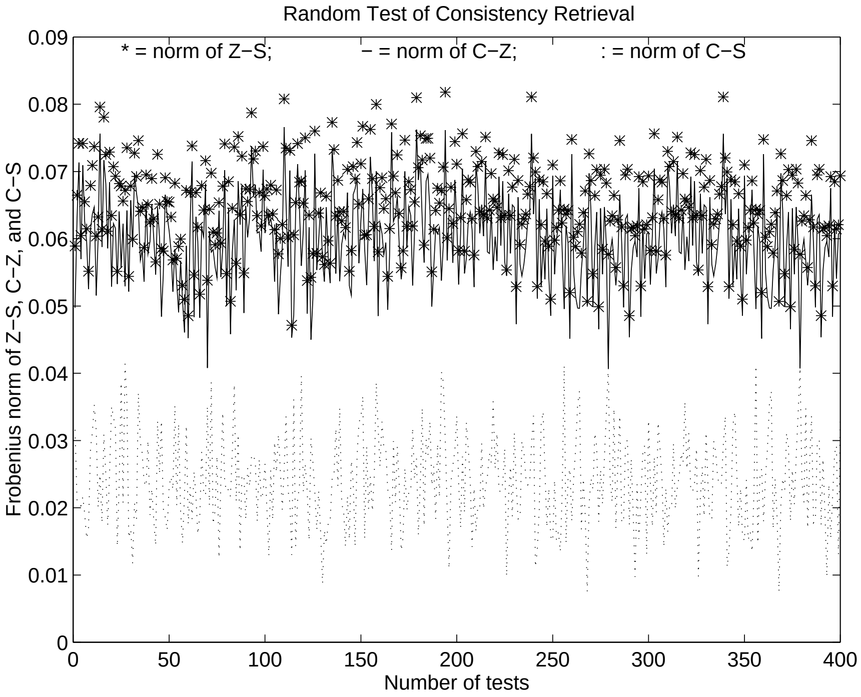

FIG. 4. *Differences of* $||Z - S||_F$, $||C - Z||_F$ *and* $||C - S||_F$ *in Example 2.*

REFERENCES

[1] D. Cox, J. Little, D. O'Shea, Using Algebraic Geometry, to be published by Springer-Verlag, draft, 1997.:w

[2] G. Crawford, The geometric mean procedure for estimating the scale of judgment matrix, Math. Modelling, 9(1987), 327-334.

[3] A. Grace, Optimization Toolbox User's Guide, The MathWorks, Inc., 1992.

[4] M. Herman, W. Koczkodaj, A Monte Carlo study of pairwise comparison, Inform. Processing Letters, 57(1996), 25-29.

[5] R. Jensen, An alternative scaling method for priorities in hierarchical structures, J. Math. Psychology, 28(1984), 317-332.

[6] T. Saaty and L. Vargas, Comparison of eigenvalue, logarithmic least square and least square methods in estimating ratios, Math. Modelling, 5(1984), 309-324.

[7] L. L. Thurstone, A law of comparative judgments, Psychological Review,. 34(1927), 273-286.