SUBMODEL SELECTION AND EVALUATION IN REGRESSION- THE X-RANDOM CASE

By

Leo Breiman
Philip Spector
Department of Statistics
University of California
Berkeley, California 94720

Technical Report No. 197
March 1989
(revised June 1990)

Statistics Department
University of California
Berkeley, California 94720

Submodel Selection and Evaluation in Regression. The X-Random Case.

Leo Breiman*
Philip Spector

University of California
Berkeley, California 94720

ABSTRACT

Often, in a regression situation with many variables, a sequence of submodels is generated containing fewer variables using such methods as stepwise addition as deletion of variables, or "best subsets". The question is which of this sequence of submodels is "best", and how can submodel performance be evaluated. This was explored in Breiman [1988] for a fixed X-design. This is a sequel exploring the case of random X-designs.

Analytical results are difficult, if not impossible. This study involved an extensive simulation. The basis of the study is the theoretical definition of prediction error (PE) as the expected squared error produced by applying any prediction equation to the distributional universe of $(y, \mathbf{x})$ values. This definition is used throughout to compare various submodels.

There are startling differences between the x-fixed and x-random situations and different PE estimates are appropriate. Non-resampling estimates such as C_p, adjusted R^2, etc. turn out to be highly biased and almost worthless methods for submodel selection. The two best methods are cross-validation and bootstrap. One surprise is that 5 fold cross-validation (leave out 20% of the data) is better at submodel selection and evaluation than leave-one-out cross-validation. There are a number of other surprises.

* Work supported by NSF Grant No. DMS-8718362.

Dans l'analyse de problèmes de régression à plusieurs variables (indépendentes), on produit souvent une série de sous-modèles constitués d'un sous-ensemble des variables par des méthodes tels que l'*addition par étope*, le *retroit par étope* et la méthode du ''meilleurs sous-ensemble''. Le problème est de déterminer liquel de ces sous-modèles est ''le meilleux et d'évaluer sa performance. Ce problème fut exploré dans Breiman [1988] dans le cas d'une matrice X fixe. Dans ce qui suit, on consideère le cas de la matrice X etant aliátoire.

La détermination de résultats analytiques est dificile, si non impossible. Hors cet(te) étude implique des simulations de grande ervergure. cet(te) étude se base sur la définition théorique de l'erreur de prédiction (PE) comme étant l'espérance du carré de l'erreur produite en applicant une équation de prédiction à l'inverse distributionel des voleurs (y,x). cette définition est utilisée afin de comparen divers sous-modèles.

La différence entre les cas de la matrice X fixe et aléatoire est remarkable et différents estimateurs du PE s'appliquent. Les estimateurs n'utilisant pas de ré-échantillonage, tels que le C_p et le R^2 ajuoté, produisent des méthodes de sélection grandement biaisées. Leo deux meilleurs méthodes sant *cross-validation* et l'auto-armarcaze *bootstrap*. Une surprise est que *5-fold cross-validation* est mieux que *leave-one-out cross-validation*. Il y a falusieurs outres résultats surprenants.

SUBMODEL SELECTION AND EVALUATION IN REGRESSION- THE X-RANDOM CASE

Leo Breiman
Philip Spector
Department of Statistics
University of California
Berkeley, California 94720

1. Introduction.

In previous research (Breiman, 1988) we explored the issue of submodel selection and evaluation when the X-design was fixed and results were conditional on the fixed X-design. In this present work we look at the situation where the X-design is random.

More specifically, we assume that there is data of the form $(y_n, \mathbf{x}_n)$, $n = 1, \ldots, N$ where $\mathbf{x}_n$ is an M-variate vector. The analyst runs a program that produces regressions based on subsets of the variables, such as a "best subsets" program or stepwise forward variable addition or stepwise backwards variable deletion. This produces a sequence of subsets $\zeta_0, \ldots, \zeta_M$ where we take each ζ_J to denote the indices of the variables in the regression and $|\zeta_j| = J$.

The problem is to select the "best" one of these submodels and to give some estimate of the predictive capability of the submodel selected. In the previous paper we set up some conceptual definitions to give some flesh to the concept of "best" and predictive capability. The basic definitions were the x-fixed and x-random prediction errors.

Suppose we have a predictor $\hat{\mu}(\mathbf{x})$ for y based on $\mathbf{x}$. In the x-fixed case, consider new data $(y_n^{new}, \mathbf{x}_n)$, $n = 1, \ldots, n$ where the y_n^{new} have the same distribution and define as the original y_n.

$$PE_c = E\|y^{new} - \hat{\mu}(\mathbf{x})\|^2$$

where we use the notation $\|a\|^2 = \Sigma a_n^2$, and the expectation is over the $\{y_n^{new}\}$ only, which are assumed independent of the original $\{y_n\}$.

If the model generating the $\{y_n\}$ is

$$y_n = \mu^*(\mathbf{x}_n) + \varepsilon_n, \quad n = 1, \ldots, N \tag{1.1}$$

with $E\varepsilon_n = 0$, $E\varepsilon_n\varepsilon_{n'} = \sigma^2\delta_{nn'}$. Then

$$PE_F = N\sigma^2 + \|\mu^* - \hat{\mu}\|^2.$$

We referred to the term $\|\mu^* - \hat{\mu}\|^2$ as the x-fixed model error ME_F.

The assumptions for the x-random case are that the (y_n, x_n) are i.i.d. sampled from the distribution of (Y, X). In this case, the predictor error was defined as

$$PE_R = N \cdot E(y^{new} - \hat{\mu}(x^{new}))^2$$

where (y^{new}, x^{new}) is a random vector with the distribution of (Y, X), but independent of the (y_n, x_n), $n = 1, \ldots, N$ and the expectation is over (y^{new}, x^{new}) only.

If Y and X are related by

$$Y = \mu^*(X) + \varepsilon,$$

$E\varepsilon = 0$, $E\varepsilon^2 = \sigma^2$, and ε independent of X, then

$$PE_R = N\sigma^2 + N \cdot E(\mu^*(x^{new}) - \hat{\mu}(x^{new}))^2.$$

The second term we called the x-random model error ME_R.

Then the submodel selection and evaluation problem was formulated as follows: let $ME(\zeta)$ be the model error for OLS regression based on the subset of variables with indices in ζ. In the sequence $\zeta_0, \ldots, \zeta_M$ find J that minimizes $ME(\zeta_J)$ and estimate $\min_J ME(\zeta_J)$.

We pointed out that what was really being selected was submodel dimensionality, since in general, there could be many different submodels of the same dimension as ζ_J with almost the same residual sum-of squares.

(1.1) X-Fixed vs X-Random:

If we assume a classical linear model then for the full M-variable model the expected x-fixed model error is $M\sigma^2$, or σ^2 per variable. In the X-random case, as we will see, the expected model error is $cM\sigma^2$, where c can be substantially larger than one.

In the latter case, more is gained by variable deletion, the "best" submodel is smaller than in the X-fixed case and has larger model error. As the sample size $\to \infty$ these differences become small. But they can be quite significant for sample sizes not large compared to the number of variables.

The model error in the X-random case reflects both the variability due to the noise components $\{\varepsilon_n\}$ and that due to the randomness in the $\{x_n\}$ as a sample from the X distribution. If M is a substantial fraction of the sample size N, the latter variability can contribute more to the ME than the former.

To illustrate this, we look at the full model ME assuming

$$\mu^*(x) = \sum_m \beta_m^* x_m$$

$$\hat{\mu}(x) = \sum \hat{\beta}_m x_m$$

where $\hat{\beta}_m$ are the OLS estimates. Let $S = X^t X$ where X is the data matrix. Then, denoting $\Gamma_{ij} = EX_i X_j$

$$ME_F = (\hat{\beta} - \beta^*) S (\hat{\beta} - \beta^*)$$

$$ME_R = (\hat{\beta} - \beta^*) N\Gamma (\hat{\beta} - \beta^*)$$

or,

$$ME_R = (\hat{\beta} - \beta^*)(N\Gamma S^{-1}) S (\hat{\beta} - \beta^*). \tag{1.2}$$

The extent to which $N\Gamma S^{-1}$ differs from the identity will govern how much ME_F and ME_R differ. One useful estimate for $N\Gamma S^{-1}$ is given by cross-validation,

$$N\Gamma S^{-1} \approx \sum_n x_n^t x_n S_{-n}^{-1}$$

where S_{-n} is the inner product matrix $X^t X$ formed with the exclusion of the n^{th} case. If we use the identity

$$S_{-n}^{-1} = S^{-1} + \frac{(S^{-1}x_n)^t (S^{-1} x_n)}{1 - h_n}$$

where $h_n = x_n S^{-1} x_n$, then we get

$$ME_R \approx ME_F + \sum_n \frac{h_n}{1 - h_n} (\mu^*(x_n) - \hat{\mu}(x_n))^2.$$

Under (1.1), taking expectations only over $\{\varepsilon_n\}$,

$$E(ME_R) \approx M\sigma^2 + \sigma^2 \cdot \sum_n \frac{h_n^2}{1 - h_n}. \tag{1.2}$$

The average h_n is $\bar{h} = M/N$. If the $\{h_n\}$ are fairly constant,

$$E(ME_R) \cong \left[\frac{N}{N-M}\right] \cdot M\sigma^2.$$

For $M = 40$ and $N = 60$, this gives $E(ME_R) \cong 3M\sigma^2$. But if the X-distribution is skewed and long tailed, some of the $\{h_n\}$ can get near one, with the result that $E(ME_R) = cM\sigma^2$, with c as high as 6-7. This will be further illustrated by our simulation results.

(1.2) *Which schema should be used?*

In some applications the x-variables are actually controlled and fixed. Here there is no question of the appropriateness of fixed x methods. But in many other situations, e.g. observational data, where there is no hope of controlling or replicating the x-variables, should PE_F or PE_R be used as the "standard"?

An interesting discussion relevant to this issue is in an article by J. Wu (1986). Referring. to the fact that unconditional confidence interval estimates need the assumption that the $\{x_n\}$ are i.i.d. samples, he states "In data analysis how often do analysts bother to find out what the sampling design is? On the other hand, a conditionally valid procedure... does not require such a stringent condition on the sampling design". In the discussion, Tibshirani refers to both conditional and unconditional procedures as being based on different "gold standards" and argues that it is not clear which one to use if the x-design is not apriori fixed.

Tibshirani's point is a good one. Much of statistics has to do with the establishing of standards for the presentation of results and for the understanding of these results.

Suppose, for example, that a faculty member has his freshman class fill out a questionaire with, say, 40 responses and then regresses the first response on the other 39. Would the x-fixed or x-random PE be a better measure of the accuracy of the results? What standard should he use?

To argue that in the absence of knowing that the x-variables are i.i.d. selected from a well-defined universe, it is better to assume they are fixed (replicable, controlled) is an argument for a poor standard. In this context, the x-random ME is a much more realistic standard.

Not only that, but if the faculty member decides to repeat the questionaire on the following years' freshman class and use the new data to estimate the prediction error of the equation derived the previous year, then his estimate is clearly much closer in concept to the x-random PE than the x-fixed.

Our belief is that for observational data, where the x-variables are gathered in an uncontrolled manner, the x-random PE is a better standard, both conceptually and also in terms of estimating prediction accuracy on future data gathered in a similar way (i.e. another freshman class).

This is a practical issue as well as a conceptual one. Methods for estimating the prediction or model error depend on whether one wishes to estimate the x-fixed or x-random values. As Efron (1986) points out, cross-validation gives an estimate of the x-random PE, and should not be used as an estimate of the x-fixed PE unless the sample size is large enough to make their difference small.

(1.3) *Outline of paper*

In the arena of submodel selection and evaluation, exact analytic results are hard to come by. Some were given in the previous paper for the x-fixed case. But the x-random case seems to be a harder nut to crack.

However, the problem is too important and pressing to be put off pending the appearance of analytical results. The standard methods used in estimating PE and selecting submodels are highly biased and usually do poor selection. Here, by standard methods we mean such things as adjusted R^2, C_p, F-to-enter, F-to-delete etc. Reviews of these appear in Miller [1984] and Thompson [1978].

Data resampling methods such as cross-validation and the bootstrap have become a hot item in this arena and are being advocated as better PE estimators and submodel selectors. However, no telling results have yet been published. For these reasons, we decided to embark on a simulation study having much of the same structure as the earlier study in Breiman (1988). It uses 40 variables at sample sizes 60 and 160.

The basic structure is this: the $\{\mathbf{x}_n\}$ are i.i.d. sampled from an underlying $\mathbf{X}$ distribution. The $\{y_n\}$ are formed from

$$y_n = \beta^* \mathbf{x}_n + \varepsilon_n, \quad \{\varepsilon_n\} \text{ i.i.d. } N(0, \sigma^2).$$

Backwards deletion of variables is used to get the sequence $\zeta_0, \ldots, \zeta_M$. The model using all M variables is called the full model.

The exact ME and PE for each submodel is computed, so we know what the best submodel is, and what its ME is. This is then compared to the ME estimates and submodels selected by a number of different procedures.

In section 2, we give an outline of the methods to be compared. Section 3 discusses the structure of the simulation. Section 4 gives the global simulation results, and section 5 the results relevant to submodel selection and evaluation. Section 6 discusses the results of some substudies, and 7 presents our conclusions.

2. **Methods to be compared.**

Denote by $\hat{\mu}(\zeta)$ the OLS predictor based on the subset of variables with indices in ζ, and let

$$ME(\zeta) = N \cdot E(\mu^*(X^{new}) - \hat{\mu}(X^{new}, \zeta))^2.$$

$$PE(\zeta) = N\sigma^2 + ME(\zeta).$$

For the particular sequence $\zeta_0, \ldots, \zeta_M$ generated by the variable selection, denote $ME(J) = ME(\zeta_J)$. Let

$$RSS(\zeta) = \|y - \hat{\mu}(\zeta)\|^2$$

and use subscript zero for full model values, i.e. $RSS_0 = RSS(\zeta_M)$. Each method given operates by forming an estimate $\hat{ME}(J)$ of $ME(J)$; selecting the submodel ζ_J such that $\hat{ME}(J) = \min_{J'} \hat{ME}(J')$ and evaluating the selected subset by its estimated model error.

(2.1) *Test set*

As a benchmark procedure, a test set $\{y_n', \mathbf{x}_n'\}$ $n = 1, \ldots, N$ is sampled, independent of the original data set, but of the same size. For any subset ζ, the test set estimate of $PE(\zeta)$ is

$$\hat{PE}(\zeta) = \Sigma (y_n' - \hat{\mu}(\mathbf{x}_n', \zeta))^2.$$

To convert this into an ME estimate an estimate of $N\sigma^2$ has to be subtracted. A reasonable σ^2 estimate is

$$\hat{\hat{\sigma}}^2 = RSS_0'/(N - M)$$

where RSS_0' is the residual-sum-of squares obtained from an OLS full model fit to the data $(y_n', \mathbf{x}_n')$.

Thus, we use as our test set ME estimate

$$\hat{ME}(\zeta) = \hat{PE}(\zeta) - N\hat{\hat{\sigma}}^2$$

which is applied to give $\hat{ME}(J) = \hat{ME}(\zeta_J)$.

(2.2) *Complete Cross-Validation*

In complete cross-validation, the nth case $(y_n, \mathbf{x}_n)$ is deleted from the data. The variable selection process is then carried out on the remaining $N - 1$ cases resulting in a sequence of subsets $\zeta_0^{(n)}$, $\zeta_1^{(n)}$,... and corresponding predictors $\{\hat{\mu}_n(x, \zeta_J^{(n)})\}$. This is done in turn for $n = 1, \ldots, N$. For each J, $J = 0, \ldots, M$, the PE estimate is

$$\hat{PE}(J) = \sum_n (y_n - \hat{\mu}_n(\mathbf{x}_n, \zeta_J^{(n)}))^2.$$

The $\hat{ME}(J)$ estimate is gotten by subtracting $N\hat{\sigma}^2$, where $\hat{\sigma}^2 = RSS_0/(N - M)$.

Complete cross-validation can be a very computer intensive process, necessitating N subset selection procedures. For this reason, we test it only at sample size 60.

(2.3) *V-fold Cross-Validation*

This procedure is a more aggregated and less expensive form of complete cross-validation. Let V be a small integer and divide the cases as nearly as possible into V equal groups. This division can be completely at random or can use some stratifying mechanism.

Denote these groups by $L_1, \ldots, L_V$ and let

$$L^{(v)} = L - L_v, \quad v = 1, \ldots, V$$

where L = all data. Using only the cases in $L^{(v)}$, do the subset selection getting the sequence $\{\zeta_J^{(v)}\}$ and predictors $\hat{\mu}_v(\mathbf{x}, \zeta_J^{(v)})$. Form the estimate

$$\hat{PE}(J) = \sum_{v} \sum_{(y_n, x_n) \in L_v} (y_n - \hat{\mu}_v(x_n, \zeta_J^{(v)}))^2,$$

and subtract $N\hat{\sigma}^2$ to get the $\hat{ME}(J)$ estimate. The initial tests of this estimate were done with V = 10.

(2.4) *Bootstrap*

The unconditional version of the bootstrap goes as follows: sample with replacement N times from the original data $\{y_n, x_n\}$. Denote the sample by $\{y_n^B, x_n^B\}$. Using the bootstrap sample, do the submodel selection getting the sequence $\{\zeta_J^B\}$ and predictors $\hat{\mu}_B(x, \zeta_J^B)$. Define

$$e_B(J) = \sum_n (y_n - \hat{\mu}_B(x_n, \zeta_J^B))^2 - \sum_n (y_n^B - \hat{\mu}_B(x_n^B, \zeta_J^B))^2$$

Then $e_B(J)$ is an estimate of the bias in RSS(J) in estimating PE(J). Repeat the bootstrap process and let $e(J) = Av_B e_B(J)$. Define the bootstrap PE estimate as

$$\hat{PE}(J) = RSS(\zeta_J) - e(J)$$

and the corresponding ME estimate by subtracting $N\hat{\sigma}^2$.

In the simulation we use 50 bootstrap repetitions. Note that we do not use the bootstrap at sample size 60. The reason is that, on the average, a bootstrap sample will omit a fraction e^{-1} of the cases. With 60 cases and 40 variables, this means that often, when the matrix X^tX is formed from the bootstrap sample, it is singular.

We could not see any method, both simple and reasonable, to get around this. A smoothed version of the bootstrap would not encounter this difficulty, but it is not at all clear how to smooth in a 40 dimensional space. Skipping any bootstrap sample where X^tX was nearly or exactly singular was another possibility, but we reasoned that this would destroy the distributional rationale for bootstrap.

(2.5) *Partial Cross-validation*

Unlike the methods above, partial cross validation only uses the main sequence of subsets $\zeta_0, \zeta_1, \ldots$ initially selected. Given any subset of variables with indices in ζ and OLS predictor $\hat{\mu}(x, \zeta)$, the cross-validated estimate for the PE is

$$\hat{PE}(\zeta) = \sum_n \left[r_n(\zeta) / 1 - h_n(\zeta) \right]^2 \qquad (2.1)$$

where the $\{r_n(\zeta)\}$ are the residuals $y_n - \hat{\mu}(x_n, \zeta)$ and $h_n(\zeta) = x_n S^{-1} x_n$, $S = X^tX$ where X^tX is formed using only the variables $\{x_m; m \in \zeta\}$. Again, $\hat{ME}(\zeta)$ is formed by subtracting $N\hat{\sigma}^2$. This equation is applied to each of the $\{\zeta_J\}$ to get the $\hat{ME}(J)$ estimates.

The idea here is based on this reasoning: in complete cross-validation, when a single case is left out, the sequence of selected subsets $\zeta_0^{(n)}, \zeta_1^{(n)}, \ldots$ should usually be identical to the sequence of subsets $\zeta_0, \ldots$ selected using the same procedure on all the data. Therefore, we can approximate complete cross validation (and drastically reduce computing time) by assuming that

$$\zeta_0^{(n)}, \zeta_1^{(n)}, \ldots \equiv \zeta_0, \zeta_1, \ldots$$

Under this assumption, complete cross-validation reduces to what we call ''partial cross-validation.''

(2.6) *Other estimates*

There are other proposed PE estimates. For instance, Mallows C_P and the S_P statistic (see Thompson [1978], Breiman and Freedman [1983]). There are also some proposed variants of V-fold cross-validation. Burman [1989] has given a first-order correction term. Stratification of the cross-validation samples has been suggested. An open issue is how many ''folds'' to use, i.e. how big should V be? An analogous question is how many bootstrap iterations should be used?

Our plan is to first give results for the test set benchmark estimate of section 2.1 and for the 4 estimates defined in 2.2 to 2.5. These latter are, to us, the current serious contenders. In section 6, we give some simulation results relevant to the other estimates.

3. **Simulation Structure.**

a) For each run, the **X**-distribution was fixed, as were the coefficients of the full model. In each repetition the x-variables were independently sampled from the underlying **X**-distribution. Normal noise was generated and added to give the y-values. Backwards deletion was then carried out to give the sequence of submodels. There were always forty variables and either 60 or 160 cases. In each run, there were 500 repetitions (with one exception noted later).

b) In each repetition the true ME was computed for each submodel selected by the backwards deletion. Various ME estimates for each submodel were derived using the methods listed in section 2.

c) Two general behavioral characteristics were observed. The first was the behavior of the ME estimates over the entire sequence of submodels. Since the true ME was known, the behavior of the estimates could be compared to it and systematic differences noted. We call this the global behavior.

The second type of behavior studied was the ability of these estimates to select submodel dimensionality and estimate the ME of the selected submodel. Knowing the true ME, we knew the optimal dimensionality.

Using each ME estimate, in each repetition we selected the submodel having the minimum test set estimated ME. For this submodel we computed its dimensionality and the value of its ME estimate. The selected dimensionality was compared with the optimal dimensionality. The ME estimate for this submodel was also compared with the true ME of the submodel. We refer to these results as the submodel selection and evaluation behavior.

d) Details: Two X-distributions were used. The first was a multivariate mean-zero normal with $E(X_iX_j) = \rho^{|i-j|}$, with $\rho = .7$. The second was a multivariate mean-zero lognormal with the same covariance matrix and coefficient of variation 1.4. In both cases $N(0, 1)$ noise was added. The non-zero coefficients were in three clusters of adjacent variables with the clusters centered at the 10^{th}, 20^{th}, and 30^{th} variables.

For the variables clustered around the 10^{th} variable, the initial coefficients values were given by

$$\beta^*_{10+j} = (h - j)^2, \quad |j| \leq h.$$

The coefficient clusters at 20 and 30 had the same shape. All other coefficients were zero. The coefficients were then multiplied by a common constant to make the theoretical R^2 equal to .75.

We used the h-values 1, 2, 3, 4. This gave, respectively, 3, 9, 15, 21 non-zero coefficients. For $h = 1$, there were three strong, virtually independent variables. At the other extreme, $h = 4$, each cluster contained 7 weak variables. These four different sets of coefficients are designated by H1, H2, H3, H4 in the tables and figures. Some t-values for the coefficients are graphed in Breiman [1988].

We also ran the case with all coefficients zero. This is designated by a Z in the tables and figures.

(3.1) *Comments on the simulation structure*

When the X-distribution is multivariate normal, the simulation is identical to that in the X-fixed case (Breiman [1988]) except that the x-variables are randomly selected in each of the 500 *repetitions* in a run, instead of being selected at the beginning of the run and held fixed.

Sampling from the multivariate normal gives relatively short tailed symmetric data distributions. The multivariate lognormal distribution is of the form

$$X_j = \alpha_j(e^{Z_j} - \beta_j)$$

with the Z_j multivariate normal, such that $EX_j = 0$, $EX_i X_j = \rho^{|i-j|}$, $\rho = .7$, and $SD(e^{Z_j})/E(e^{Z_j}) = 1.4$.

This lognormal distribution is skewed and long tailed. A few high leverage cases in each repetition is a normal occurrence. The effects of the randomness of the x-sample using this distribution are very marked.

The X-fixed simulation was run on sample sizes of 60,160, 600, and required many hours of CRAY cpu time. The X-random simulation required even more intensive computations. To keep the computing requirements within bounds, we eliminated the sample size 600 runs.

4. **Global Results.**

The best way to understand the global behavior of the estimates is to look at the graphs in Figures 1-4. The graphs on the left side of the page are the average of the various ME(J) estimates over the 500 repetitions in a run plotted as a function of J. The solid line is the average of the true ME(J).

The graphs on the right side of the page are the RMS differences between the various ME(J) estimates and the true ME(J) computed and averaged over the 500 repetitions and plotted against J. The solid line is the standard deviation of the true ME(J).

The most immediately striking result is the increase in ME over the X-fixed case. In that case, the average full model ME was close to 40 ($\sigma^2 = 1$). Here, for N = 60, the full model MEs are around 120 in the multivariate normal case and above 300 in the lognormal. The effect is less pronounced at N = 160, but the lognormal ME is still almost 100 at J = 40.

Another striking effect is the decrease in ME achieved by going from the full model to the minimum ME model. One wins big in the X-random case by going to small submodels.

Looking at the global behavior of the various estimates, we pick out the following features

i). Complete cross-validation has uniformly low bias and RMS error

ii). At N = 60, ten fold cross-validation is biased upwards with larger RMS error at the higher dimensional submodels. This bias is considerably reduced at N = 160.

iii). Bootstrap has fairly low bias and RMS error at N = 160, with the estimate tending to be slightly low.

iv). Partial cross-validation is heavily biased downward with generally high RMS error.

To get an overall measure of bias for each J we computed percent difference of the average ME(J) estimate from the average true ME(J), took the absolute value of this percent difference, and averaged over J. These numbers are given in Table 4.1.

Note that the test set procedure has theoretical bias zero. Thus, the extent to which the TS bias percentages differ from zero gives an indication of the error due to finite sample size. All of the results in this and the next section for the lognormal case, N = 60, are based on runs of 1000, instead of 500. This is the most variable situation and we decided on the basis of an initial run that the larger run length should be used.

Table 4.1
Average Percent Bias

	Z*	H1	H2	H3	H4
		Normal	N = 60		
TS	1.0	3.5	2.3	.8	3.7
CCV	4.5	4.6	2.8	3.1	2.1
CV/10	30.9	36.2	29.6	29.6	29.1
PCV	93.9	81.2	77.5	76.3	76.5
		Lognormal	N = 60		
TS	2.9	3.4	4.2	1.0	4.8
CCV	2.3	3.4	4.1	3.5	7.4
CV/10	32.3	40.9	28.1	36.4	36.9
PCV	83.4	76.3	74.2	73.1	72.4
		Normal	N = 160		
TS	1.3	2.2	.5	2.1	1.3
BOOT	25.9	24.2	20.4	16.0	19.1
CV/10	12.5	12.3	12.5	15.2	12.3
PCV	84.8	72.0	65.0	61.1	60.8
		Lognormal	N = 160		
TS	2.3	2.3	1.4	1.8	1.2
BOOT	15.4	14.6	10.3	8.0	12.8
CV/10	15.9	17.4	14.7	19.4	13.8
PCV	78.0	67.8	64.7	61.2	61.7

*Averaged only over $J \geq 4$.

The overall measure of RMS difference is calculated as follows: in each run, look at those J values for which the average true ME(J) is less than the average true full model ME. Now average the RMS differences between the true and estimated ME(J) over this set of J values. This partial averaging eliminates a few of the smallest J values for which the ME(J) values and corresponding RMS values are very large. These results are given in table 4.2.

Table 4.2
Average RMS Error

	Z	H1	H2	H3	H4
		Normal	N = 60		
TS	28.7	28.6	30.9	31.3	30.6
CCV	43.3	48.9	50.8	50.4	53.5
CV/10	59.1	65.9	65.9	65.7	71.5
PCV	75.6	75.7	82.5	83.2	85.3
		Lognormal	N = 60		
TS	238.6	142.6	137.0	236.4	151.2
CCV	169.9	196.1	240.8	233.3	227.1
CV/10	217.6	258.7	354.1	272.7	353.8
PCV	193.2	206.7	212.6	226.8	217.7
		Normal	N = 160		
TS	17.3	16.3	18.2	18.6	19.4
BOOT	15.8	16.4	16.8	16.8	18.8
CV/10	16.6	17.8	19.3	20.7	21.5
PCV	34.6	32.6	33.1	33.5	37.7
		Lognormal	N = 160		
TS	39.1	34.6	36.7	39.9	44.2
BOOT	32.8	32.2	32.2	34.9	37.4
CV/10	44.4	48.7	46.9	52.4	55.9
PCV	56.8	55.8	57.7	59.5	65.6

5. **Selection and Evaluation Behavior.**

The most important role of the PE/ME estimates is in submodel selection and evaluation; i.e. how good a submodel does it select and how good is the ME estimate of the

selected submodel.

To answer the question of how good the selection is, the criterion used is the average true ME value for the selected submodel. This is given in Table 5.1.

The next comparison is between the average dimension as selected by the true ME and by each of the estimates, together with the RMS differences between them. This is given in Table 5.2 where the numbers in parentheses are the RMS differences, except that the number following the average dimension selected by the true ME is the standard deviation.

Table 5.1
True ME of Submodels Selected

	Z	H1	H2	H3	H4
		Normal	N = 60		
ME(True)	0	7.9	20.5	31.9	37.8
TS	1.4	9.6	23.0	35.1	41.9
CCV	5.8	16.5	30.9	46.8	55.1
CV/10	2.7	13.4	28.8	43.0	51.2
PCV	56.2	62.4	78.8	83.0	83.2
		Lognormal	N = 60		
ME(True)	0	31.3	42.8	52.4	57.4
TS	1.7	36.7	51.4	60.3	67.9
CCV	8.3	52.7	73.8	86.1	92.2
CV/10	5.1	52.7	66.1	80.6	86.0
PCV	108.3	149.1	168.2	174.6	166.8
		Normal	N = 160		
ME(True)	0	3.0	18.2	28.6	35.3
TS	1.3	4.2	21.0	32.0	38.8
BOOT	1.7	5.6	26.3	41.7	49.1
CV/10	3.4	7.9	24.5	40.0	49.3
PCV	29.1	30.2	38.0	40.9	46.2
		Lognormal	N = 160		
ME(True)	0	4.1	23.1	41.7	52.3
TS	1.2	5.5	26.6	47.3	59.2
BOOT	1.5	7.0	31.2	57.5	70.8
CV/10	3.6	8.6	31.7	59.5	72.7
PCV	44.6	45.9	57.1	66.7	73.9

Table 5.2

Dimension Selected

	Z	H1	H2	H3	H4
			Normal N = 60		
ME(True)	0.0(0.0)	3.2(.7)	4.1(1.3)	4.5(1.9)	5.5(2.7)
TS	.9(1.1)	4.1(1.5)	4.3(2.6)	5.1(3.0)	6.1(4.2)
CCV	1.9(2.3)	5.6(3.5)	1.7(3.8)	5.9(5.0)	6.7(5.8)
CV/10	.8(1.1)	4.4(2.7)	5.5(2.8)	4.7(3.4)	5.5(4.2)
PCV	11.1(12.3)	12.9(11.1)	14.6(12.1)	15.1(11.7)	15.1(11.0)
			Lognormal N = 60		
ME(True)	0.0(0.0)	3.2(.9)	3.6(1.2)	4.1(1.5)	4.6(1.7)
TS	.4(.9)	3.7(2.7)	4.3(3.1)	4.6(3.2)	5.2(3.7)
CCV	.6(1.7)	3.9(2.9)	4.5(3.9)	5.0(3.7)	5.3(4.6)
CV/10	.4(1.0)	3.4(1.9)	4.0(3.0)	4.3(2.9)	4.6(3.6)
PCV	9.9(11.4)	12.6(5.7)	13.4(11.3)	14.0(5.3)	13.6(10.3)
			Normal N = 160		
ME(True)	0.0(0.0)	3.0(0.0)	4.2(1.6)	3.6(3.2)	10.8(4.3)
TS	.3(.9)	3.3(.9)	5.3(4.0)	9.3(5.2)	12.2(6.8)
BOOT	.2(.7)	3.3(1.0)	5.9(5.2)	2.7(10.5)	17.6(13.6)
CV/10	.5(1.3)	3.7(1.7)	5.1(4.0)	1.1(6.7)	12.1(8.9)
PCV	7.7(8.4)	9.8(7.5)	12.0(8.5)	13.4(6.5)	14.2(6.4)
			Lognormal N = 160		
ME(True)	0.0(0.0)	3.0(.1)	4.0(1.6)	7.3(3.4)	9.2(4.0)
TS	.3(.7)	3.3(.9)	4.8(3.1)	8.8(6.1)	11.3(7.6)
BOOT	.1(.4)	3.2(.8)	4.5(2.7)	7.3(5.7)	9.3(6.8)
CV/10	.4(1.1)	3.5(1.5)	4.8(3.5)	8.1(7.0)	10.1(8.0)
PCV	7.9(8.8)	10.1(8.0)	11.6(8.6)	12.9(7.6)	13.7(7.2)

In terms of the ability of the estimate to evaluate the subset selected, we give two tables. The first (Table 5.3) compares the average estimated ME value for the subset selected by the estimate to the average true ME value for the same subset. In this table, the numbers in parentheses are the true ME averages. Table 5.4 gives the RMS differences between the true ME and the estimated ME for the subset selected over the 500 repetitions in a run (1000 for lognormal n = 60).

Table 5.3

Average Estimated ME

	Z	H1	H2	H3	H4
			Normal N = 60		
TS	-1.2(1.4)	6.0(9.6)	19.7(23.0)	28.7(35.1)	35.0(41.9)
CCV	-2.7(5.8)	4.9(16.5)	18.1(30.9)	28.0(46.8)	31.4(55.1)
CV/10	-1.9(2.7)	10.6(13.4)	24.9(28.8)	37.1(43.0)	41.5(51.2)
PCV	-13.4(56.2)	-10.6(62.4)	-8.4(78.8)	-7.6(83.0)	-8.0(83.2)
			Lognormal N = 60		
TS	-.1(1.7)	25.1(36.7)	38.9(51.4)	44.4(60.3)	49.9(67.9)
CCV	-1.7(8.3)	25.0(52.7)	42.7(73.8)	46.2(86.1)	47.0(92.2)
CV/10	-1.3(5.1)	35.5(52.7)	47.8(66.1)	59.3(80.6)	58.2(86.0)
PCV	-12.0(108.2)	-8.7(149.1)	-6.3(168.2)	-6.2(174.6)	-6.5(166.8)
			Normal N = 160		
TS	-.1(1.3)	2.3(4.2)	15.2(21.0)	25.8(32.0)	32.1(38.8)
BOOT	.3(1.7)	2.9(5.6)	17.9(26.3)	29.0(41.7)	32.7(49.1)
CV/10	-.7(3.4)	2.0(7.9)	14.4(24.5)	27.3(40.0)	35.4(49.3)
PCV	-10.2(29.2)	-6.9(30.2)	-3.7(38.0)	-1.5(40.9)	-.4(46.2)
			Lognormal N = 160		
TS	-1.0(1.2)	3.0(5.5)	19.3(26.6)	34.0(47.3)	45.2(59.2)
BOOT	.3(1.5)	8.0(7.0)	26.4(31.2)	42.8(57.5)	49.9(70.8)
CV/10	-.5(3.6)	4.5(8.6)	21.0(31.7)	40.6(59.5)	51.2(72.7)
PCV	-10.4(44.6)	-7.0(45.9)	-3.5(57.1)	-.5(66.7)	0(73.9)

Table 5.4
RMS Differences in MEs

	Z	H1	H2	H3	H4
		Normal	N = 60		
TS	16.0	16.9	20.4	21.7	22.5
CCV	20.0	27.5	31.0	36.1	40.7
CV/10	16.4	23.0	26.3	26.9	30.3
PCV	76.7	82.7	97.7	98.0	81.5
		Lognormal	N = 60		
TS	16.1	31.8	42.8	42.8	41.8
CCV	27.7	61.3	119.7	78.7	83.2
CV/10	21.7	59.3	62.2	68.5	64.2
PCV	151.3	193.7	210.1	212.4	201.5
		Normal	N = 160		
TS	10.8	10.7	15.0	16.4	17.6
BOOT	9.7	10.1	15.1	19.9	22.6
CV/10	11.8	13.6	16.9	21.4	24.6
PCV	41.1	39.0	43.2	44.3	48.7
		Lognormal	N = 160		
TS	11.0	11.9	20.0	26.0	32.5
BOOT	10.9	14.1	20.5	30.7	36.3
CV/10	14.1	17.4	26.1	36.9	41.3
PCV	60.4	58.5	66.3	72.8	78.7

The major surprise here is that ten-fold cross-validation is uniformly better in selection/evaluation than complete cross-validation. Complete cross validation has better global behavior. But the critical issue in selection is the shape of the estimates ME(J) curve near the minimum value of the true ME(J) curve, rather than global behavior. Where it counts CV/10 gives better performance than CCV.

At sample size 160, CV/10 and bootstrap give very competitive results. In selection, there is very little to choose between them. In evaluation, bootstrap has a slight edge.

Partial cross validation's performance is not in the same league. It is so poor that it should not be seriously considered for submodel selection/evaluation in regression.

6. Some Substudies.

In the studies discussed below, the summary statistics for some of the estimates may differ somewhat from the same summary statistics for the same estimates given in the previous sections. This is because different random numbers may have been used. But whenever two or more procedures are compared below, the comparison is on runs on the same data.

(6.1) *Fixed path estimates*

By *fixed path* estimates of ME/PE we mean estimation methods that work with the given sequence $\zeta_0, \ldots, \zeta_M$ of submodels only. For example, partial cross validation is a fixed path estimate. But 10-fold cross-validation generates 10 generally different sequences of submodels in addition to the initial sequence. In contrast, we refer to estimates that generate other sequences of submodels as *alternative path* estimates.

Partial cross-validation is the most complicated of the fixed path estimators. Others in common use are the C_P estimate of PE (ζ_J) given by,

$$\mathrm{RSS}(\zeta_J) + 2\hat{\sigma}^2 J$$

the S_P estimate given by (approximately)

$$\left[\frac{N}{N-J}\right]^2 \mathrm{RSS}(\zeta_J)$$

Various asymptotic optimality properties can be given for some of these estimates, if no extensive data driven submodel selection is used.

But in realistic situations, such as in the structure of this simulation, fixed path estimates are hopelessly biased and do poorly in subset selection. This was the case for C_P in the X-fixed study and for partial cross-validation in the present study.

We also calculated and used C_P in the present study. The results were similar to those using partial cross-validation, but C_P gave somewhat better results. For example, Table 6.1 compares the true ME values for the subsets selected by PCV and CP for $N = 60$.

Table 6.1
True ME Values

	Z	H1	H2	H3	H4
			Normal		
CP	35	45	64	70	72
PCV	56	62	79	83	83
			Lognormal		
CP	79	119	146	149	145
PCV	108	149	168	175	167

Comparing these results with the alternative path estimates (Table 5.1) shows that the improvement is not of much help. Figure 5 gives the graphs of the global C_P behavior compared to PCV for the normal distribution, N = 60.

We did not run the S_P estimate. One reason is that, at least in the normal case, it should be close to the partial cross-validation value. Looking at the definition of the latter, note that if the $h_n(\zeta_J)$ are almost constant, then since $\sum_n h_n(\zeta_J) = J$, we can approximate the $1/(1 - h_n(\zeta_J))^2$ term in (2.1) by $N^2/(N - J)^2$. this gives the corrected residual-sum-of-squares estimate

$$\hat{PE}(J) = (N/N - J)^2 RSS(\zeta_J)$$

and the associated $\hat{ME}(J)$ estimate. This is very close to the S_P statistic recommended by Thompson [1978] in her review article. For an asymptotic justification of S_P see Breiman and Freedman [1983].

(6.2) *Correcting and stratifying the CV/10 estimate*

Burman [1989] gives a first order correction to the V-fold CV estimate of PE. Another issue in this estimation method is how to select the V subsets into which the N cases are grouped. The simplest method is ordinary random selection. But the question has been raised as to whether some sort of stratified selection might improve accuracy.

In particular, in the lognormal x-distribution, a few very high leverage cases usually occurred in the full model. Thus, sampling from strata determined by the leverage values (diagonals of the hat matrix) in the full model, might give a more homogeneous grouping and increased stability. More specifically, the data were sorted by their full model h_n values and divided into N/V groups. One observation from each of these

groups was then randomly selected (without replacement) to form each of the $L_1, \ldots, L_V$.

For the normal case, N = 60 figure 6 gives plots of the global behavior of the estimate (CV/C) resulting from correcting CV/10, the estimate (CV/S) resulting from stratifying and then doing 10 fold cross-validation, and the estimate (CV/CS) resulting from both correcting and stratifying. The correction does improve accuracy for the larger submodels. It is not clear that the stratification has any effect.

However, the story in subset selection and evaluation indicates that neither the correction or stratification are useful. For instance, Table 6.2 gives the true average ME for the submodels selected by the different estimates for sample size 60.

Table 6.2
True ME Values

	Z	H1	H2	H3	H4
		Normal	N = 60		
CV/10	2.7	13.4	28.8	43.0	51.2
CV/C	3.6	17.4	33.5	46.4	54.4
CV/S	3.2	13.2	28.8	44.1	52.7
CV/CS	4.5	16.6	33.5	46.6	55.3
		Lognormal	N = 60		
CV/10	5.1	52.7	66.1	80.6	86.0
CV/C	9.1	60.0	72.5	85.4	92.9
CV/S	5.4	52.7	69.0	81.0	84.6
CV/CS	11.5	58.4	74.4	90.1	94.0

The thought might occur that even if CV/C did not do as well in submodel selection, it might be a better ME estimator at the subset it selects. Not so! In every case the corrected estimate does worse than the uncorrected estimate.

Thus, using the correction term makes selection and evaluation less accurate. Our method of stratification seemed to neither help or harm.

(6.3) *How many folds in cross-validation?*

The preceding sections have produced some surprises concerning cross-validation. Ten fold validation gave better selection and evaluation results than complete cross-validation, even though the latter is a better global estimate. Similarly, adding a correction term gives a better global estimate, but a poorer selection/evaluation

method. This raises the possibility that 5-fold or even 2-fold cross-validation estimates might be reasonably good in selection/evaluation. For N = 60, 2-fold was not possible, leading to a singular X^tX matrix.

Thus, we compared CV/10 to CV/5 at N = 60 and CV/10, CV/5 and CV/2 at N = 160. The global results are as expected: CV/5 and CV/2 have larger bias and RMS error at the larger submodels (see figure 7,8 for graphs in the normal case.) To compare the selection/evaluation performance, we created Table 6.3 and 6.4. Table 6.3 gives the true average ME for the selected subset, and 6.4 gives the RMS error for the ME estimate of the selected subset.

Table 6.3

True ME

	Z	H1	H2	H3	H4
		Normal	N = 60		
CV/10	2.4	12.3	30.8	42.5	53.4
CV/5	1.7	11.5	28.9	41.0	53.9
		Lognormal	N = 60		
CV/10	3.2	49.4	65.7	81.9	88.3
CV/5	2.6	52.6	70.4	81.7	87.6
		Normal	N = 160		
CV/10	3.0	6.2	24.4	40.9	47.1
CV/5	2.2	5.2	23.3	41.3	48.0
CV/2	1.2	3.5	22.2	43.3	54.9
		Lognormal	N = 160		
CV/10	3.7	9.6	33.6	58.6	71.4
CV/5	2.2	8.0	32.4	57.9	71.7
CV/2	1.1	9.0	32.2	62.4	82.2

Table 6.4
RMS Error

	Z	H1	H2	H3	H4
		Normal	N = 60		
CV/10	16.3	12.4	30.8	42.5	53.4
CV/5	15.8	11.5	28.9	41.0	53.9
		Lognormal	N = 60		
CV/10	17.6	65.4	56.5	66.1	65.4
CV/5	16.8	65.7	62.1	61.1	62.9
		Normal	N = 160		
CV/10	11.9	12.8	16.8	20.9	22.9
CV/5	10.9	11.7	14.6	19.2	22.4
CV/2	10.0	14.8	23.4	30.9	36.2
		Lognormal	N = 160		
CV/10	14.1	19.6	28.1	37.7	40.4
CV/5	12.1	19.2	28.1	36.0	40.8
CV/2	10.7	68.7	54.9	63.1	69.9

We see again the interesting phenomenon that although CV/5 is not as good an estimator *globally* as CV/10, it does as well on submodel selection and evaluation. But two folds are not enough and tables 6.3 and 6.4 show CV/2 breaking down in accuracy.

The breakdown of CV/2 seems to have its source in that with a sample size of only 80, CV/2 tends to select models that are too small.

(6.4) *How many bootstraps are needed?*

In our main simulation, we used 50 bootstrap iterations. The question of how much this can be reduced without significant loss in accuracy is an important practical issue. Fifty bootstrap iterations is a considerable amount of computing (see the next section).

Table 6.5
True Average ME

	Z	H1	H2	H3	H4
		Normal	N = 160		
BOOT/50	2.3	5.1	26.6	41.0	47.9
BOOT/20	2.6	5.1	27.0	41.4	48.1
BOOT/10	2.9	5.4	27.1	41.5	48.5
BOOT/5	3.9	6.2	27.6	41.8	48.3
		Lognormal	N = 160		
BOOT/50	1.5	6.9	32.9	58.4	67.7
BOOT/20	2.1	7.4	33.1	59.3	68.4
BOOT/10	2.6	8.0	33.8	59.6	69.2
BOOT/5	3.4	9.1	35.1	60.2	69.9

To look at this issue we ran the sample size 160 cases using 50, 20, 10 and 5 bootstrap iterations (see figure 9,10). Tables 6.5 and 6.6 compares the selection/evaluation performance. Table 6.5 gives the true average ME for the selected subset and 6.6 gives the RMS estimate error for the ME estimate of the selected subset.

The accuracy of BOOT holds up even with a sharply reduced number of bootstrap iterations. Globally, there is no increase in bias and the RMS error only shows appreciable increases at 5 iterations.

The submodel selection and evaluation accuracy holds up even for as few as 5 bootstraps. The differences between 50 and 20 are small, and dropping even lower creates few ripples. Past 10-20 bootstrap iterations, the increase in accuracy is marginal compared to the computing time required.

Table 6.6
RMS Error

	Z	H1	H2	H3	H4
		Normal	N = 160		
BOOT/50	10.9	10.9	15.0	19.1	24.1
BOOT/20	11.6	11.1	15.8	19.0	23.1
BOOT/10	12.6	11.9	16.8	20.1	21.7
BOOT/5	14.1	13.6	18.9	22.1	20.8
		Lognormal	N = 160		
BOOT/50	11.0	13.8	22.9	30.9	34.0
BOOT/20	11.9	15.4	24.2	32.9	35.5
BOOT/10	13.4	15.8	25.4	34.8	37.6
BOOT/5	15.3	17.1	27.5	37.8	40.9

(6.5) *Restriction to cost-effective submodels*

The idea of cost-effective submodels was introduced in Breiman [1988], [1989] and is similar to the notion of cost-complexity submodels used in regression and classification trees (Breiman et. al. [1985]).

Briefly, given a sequence $\zeta_0, \ldots, \zeta_M$, call ζ_J a *cost minimizer* if there is an $\alpha \geq 0$ such that J minimizes $RSS(\zeta_{J'}) + \alpha J'$, $0 \leq J' \leq M$. Call ζ_J *cost effective* if it is a cost minimizer and some value of α for which it is a cost minimizer is between $2\hat{\sigma}^2$ and $10\hat{\sigma}^2$. ($\hat{\sigma}^2$ the full model estimate).

In the x-fixed simulation, the results indicated that restricting the submodel selected to be cost effective had a uniformly beneficial effect on the selection/evaluation procedure. We conducted a similar study in the present X-random situation.

Let $J_1, \ldots, J_K$ be the dimensions of the cost effective submodels. Usually, there are only a few such submodels. In fact, for all runs in the simulation, out of 41 submodels, on the average about 5 are cost-effective. Now, for any $\hat{ME}(J)$ estimate, select that $J \in \{J_1, \ldots, J_K\}$ which minimizes $\hat{ME}(J)$.

The effects of restricting choice to cost effective submodels was carried out in a separate simulation. To keep computing time down, we explored only its effect on CV/10 and BOOT results and summarized in table 6.7, 6.8 and 6.9. Table 6.7 compares the true ME of the selected subsets. Table 6.8 compares the RMS ME estimate errors, and Table 6.9 gives the average dimension selected and its RMS difference

from that selected by the true ME.

Table 6.7

True ME

	Z	H1	H2	H3	H4
		Normal	N = 60		
CV/10	3.4	12.4	28.9	41.6	52.2
CV/10/CE	3.2	12.4	28.5	41.8	50.7
		Lognormal	N = 60		
CV/10	3.3	48.2	66.6	79.7	85.4
CV/10/CE	3.1	47.5	66.3	77.7	85.8
		Normal	N = 160		
CV/10	3.7	6.3	25.2	41.2	48.6
CV/10/CE	3.6	6.3	25.1	39.7	47.3
BOOT	1.6	4.4	27.2	43.5	48.8
BOOT/CE	1.6	4.4	26.3	40.4	46.3
		Lognormal	N = 160		
CV/10	4.0	8.5	33.1	58.9	71.2
CV/10/CE	3.5	8.7	31.9	55.5	68.0
BOOT	1.6	6.7	31.9	56.6	68.8
BOOT/CE	1.6	6.9	31.9	55.9	67.8

Table 6.8

RMS Error

	Z	H1	H2	H3	H4
		Normal	N = 60		
CV/10	17.7	21.9	26.4	29.3	33.5
CV/10/CE	17.9	20.8	26.0	28.4	32.3
		Lognormal	N = 60		
CV/10	17.1	50.2	59.4	79.1	64.1
CV/10/CE	17.0	55.4	66.3	86.7	64.2
		Normal	N = 160		
CV/10	13.0	12.7	17.6	22.0	23.3
CV/10/CE	13.0	12.7	17.3	21.4	22.9
BOOT	10.4	10.7	16.7	21.1	22.5
BOOT/CE	10.4	10.7	15.3	18.3	19.4
		Lognormal	N = 160		
CV/10	15.6	16.8	26.8	35.5	40.3
CV/10/CE	14.1	16.4	26.4	38.8	42.4
BOOT	11.0	13.8	20.8	27.7	34.7
BOOT/CE	11.0	13.6	22.2	28.6	33.3

Table 6.9
Dimension Selected

	Z	H1	H2	H3	H4
			Normal N = 60		
ME(True)	.0(.0)	3.1(.5)	3.5(1.1)	4.5(1.8)	4.5(1.8)
CV/10	.5(1.7)	3.6(1.9)	4.2(2.8)	5.0(4.0)	5.7(5.1)
CV/10/CE	.4(1.7)	3.6(1.5)	4.2(1.9)	5.2(3.0)	5.6(2.7)
			Lognormal N = 60		
ME(True)	.0(.0)	3.2(.9)	3.6(1.2)	4.1(1.4)	4.5(1.7)
CV/10	.3(.7)	3.5(2.1)	4.0(3.0)	4.2(3.6)	4.3(3.1)
CV/10/CE	.2(.7)	3.7(1.7)	4.1(2.4)	4.3(2.6)	4.5(2.7)
			Normal N = 160		
ME(True)	.0(.0)	3.0(.0)	4.2(1.7)	8.6(3.0)	11.3(4.9)
CV/10	.6(1.4)	3.5(1.2)	5.1(4.1)	9.0(7.1)	12.5(9.6)
CV/10/CE	.5(1.4)	3.5(1.1)	5.1(3.1)	7.7(4.1)	9.9(4.3)
BOOT	.2(.6)	3.1(.5)	6.5(6.9)	13.5(11.8)	17.0(12.7)
BOOT/CE	.2(.6)	3.1(.4)	5.8(4.0)	10.0(5.1)	12.3(4.9)
			Lognormal N = 160		
ME(True)	.0(.0)	3.0(.1)	4.0(1.6)	7.2(2.9)	9.5(4.5)
CV/10	.4(1.5)	3.5(1.5)	4.7(3.7)	7.8(6.9)	9.9(8.3)
CV/10/CE	.3(1.0)	3.4(1.2)	4.6(2.4)	6.9(3.6)	8.4(4.1)
BOOT	.1(.4)	3.2(.7)	4.5(2.8)	7.1(5.1)	8.9(6.9)
BOOT/CE	.1(.4)	3.2(.7)	4.4(2.3)	6.8(3.6)	8.4(3.8)

These results show that selection/evaluation is about as good, and often slightly better by insisting that the submodel selected be cost effective. Table 6.9 shows, in particular, that the restriction has a stabilizing effect on the dimensionality selection.

(6.7) *Computational aspects.*

There are two interesting computational aspects we ran across in this work. The first was that after using about 50 hours of CRAY XMP-2 cpu time, we realized that we were only about half way through the simulation and almost out of CRAY money.

The rest of the simulation was done on 25 networked SUN 3/50's in the Statistical Computing Facility at the U.C. Berkeley Statistics Department. Each run of 500 iterations was split into 25 runs. The compiled code for this smaller run using a random

number as a seed to the simulation's random number generator was executed in parallel on each SUN 3/50 and the individual output files sent to the ''mother'' file system for processing.

The programs were run on low priority to avoid conflict with the normal interactive SUN usage. Since these machines are rarely used from late at night to early in the morning, the simulation had virtually exclusive use of them for 10 hours a day. Our estimate is that 25 SUN 3/50's are about 1/4 of a CRAY XMP-2. But because we did not have to wait in a queue with other CRAY users, our turn-around time was usually at least as good.

Another issue of practical importance is computational efficiency of the various estimation procedures. The fixed path procedures are most efficient but also least useful. The two most effective estimates are CV/V and BOOT. In addition to the original regression and submodel sequence generation, CV/V and BOOT do additional regressions and submodel sequence generation. In each such replicate the operations necessary consist of two main components. The first is in the formation of the X^tX matrix, where about NM^2 operations are needed. The second is in generating the sequence of submodels. If simple stepwise variable deletion or addition is used and implemented by Gaussian sweeps, then about $2M^3$ operations are used. Many more are required if a best subsets algorithm is used.

After the additional regressions have been done, they have to be combined to give the ME(J) estimates for each J. If R is the number of bootstraps or the number of folds, then this costs about $3/2\,M^3R$ operations. In CV/V the X^tX computing can be reduced. Take $\{n'\}$ to be a random permutation of $\{1, \ldots, N\}$. Let

$$X^tX_{ij}^{(v)} = \sum_{N_v \le n' \le N_{v+1}} x_{in'} x_{jn'}$$

where $N_v = [N(v-1)/V]$. Then $X^tX = \sum_v X^tX^{(v)}$, and the sum-of-squares matrix with the v^{th} group deleted is $X^tX - X^tX^{(v)}$. This reduces *all* sum-of-squares computations in CV/V from NM^2V operations to about NM^2 operations.

Another place where computation can be reduced is in the restriction to cost effective submodels. The number of operations needed to compute the ME(J) estimate is $3/2\,M^2R$ per submodel for bootstrap and CV/V respectively. If these estimates are computed only for the cost effective submodels, then the operations required in forming estimates drop by the proportion of non-cost effective submodels.

Here are some typical SUN 3/50 timing runs (CPU seconds) in cell H_3 of the simulation:

	Regular	Cost Effective
CV/5	29.0	22.5
CV/10	43.5	36.5
BOOT/5	77.2	48.2
BOOT/10	146.2	88.7
BOOT/50	698.0	413.0

The time for a single sequence of variable deletions is 7.7 CPU seconds.

7. Conclusions

(7.1) *Submodels in x-random v.s. x-fixed.*

The full model x-random ME has an expectation of about 120 for the sample size 60 simulated normal data. In the x-fixed case it is 40. For the H3 coefficients the true ME minimum submodels have an average ME of 31.9 (Table 5.1). This is 26% of the full model ME.

In the x-fixed case for the same coefficients, the similarly selected submodels had average MEs of 54% of the full model ME. This is typical across Z,H1,H2,H3,H4. In the x-random case submodel selection results in a much larger reduction of full model ME than in the X-fixed case.

This is not so pronounced for normal data at N = 160. Here, the submodel selected in the x-random setting under H3 is 54% of the full model ME compared to 62% for the x-fixed case.

The reduction can be even more drastic if the X-distribution is skewed and long-tailed. In the lognormal N = 60 case, with H3 coefficients, the full model ME is reduced to 15% percent of its value at the minimum ME submodel. The message is clear: *You can win big by using submodel selection in the x-random case*, especially for thin sample sizes and irregular X-distribution.

Another thing that shows up is that we win more by selecting smaller submodels in the x-random case. For instance, here is a comparison of the average dimension selected in the N = 60, normal runs using true ME for selection:

	Z	H1	H2	H3	H4
x-fixed	.0	3.2	4.1	6.1	7.9
x-random	.0	3.2	4.1	4.5	5.5

In the H3, H4 coefficients there are a number of weak variables. In the x-random situation there is more incentive to peel these off and reduce x-variability than in x-fixed. There is still evidence of this effect at N = 160, but not as strongly. For x-fixed the average dimension in H4 is 11.6. For x-random it is 10.8.

(7.2) *Which ME/PE estimator to use?*

We hope this present simulation will drive another nail into the practice of using fixed path estimators when data driven submodel selection is in operation.

Surprisingly, CV/V for V as low as 5 does better selection/evaluation than complete cross-validation. Bootstrap, when the sample size is large enough to use it, does as well as CV/V in selection with a small edge in evaluation, and accuracy is not significantly decreased with as few as 5 bootstrap iterations. On the other side of the scale is bootstrap's computational expense compared with CV/V.

But no matter which method is used, it seems fairly clear that restricting attention to the small class of cost effective submodels has a number of advantages and no apparent disadvantages.

(7.3) *Submodel evaluation*

All estimates of ME for the selected submodels had appreciable downward bias (see Table 5.3). But, in general, this bias was not the major factor in their RMS error (see Table 5.4). In comparing the RMS errors of all estimates (including test set) to the average ME being estimated (Table 5.1), one is disappointed by how large the RMSE/ME ratio is.

Often the RMSE is about the same size as the ME it is trying to estimate. At best it is not less than about half of the ME. This persists even as sample is increased to 160. If $ME \ll N\sigma^2$, the ME term makes a small contribution to PE and the major variation in estimating PE is in the estimation of $N\sigma^2$. This latter quantity can be estimated with small coefficient of variation for $N - M \gg 1$. In fact, some approximate calculations indicate that the coefficient of variation for estimating PE in the normal case for the subsets selected by either CV/V or BOOT is around .15 for N = 160 but over .3 for N = 60.

The reason for the noisiness in the ME estimates was discussed in Breiman [1988]. It is intrinsic in the nature of the problem. There is some evidence that using these estimates to compare submodels is more reliable. That is, given two submodels with indices in ζ_1, ζ_2, it seems possible to estimate $\mathrm{ME}(\zeta_1) - \mathrm{ME}(\zeta_2)$ with much less variability than either of $\mathrm{ME}(\zeta_1)$, $\mathrm{ME}(\zeta_2)$ separately. Thus, using bootstrap or cross-validation to compare procedures operating on the same data may give reasonable results. But answers to these issues aren't known yet.

References

Breiman, L. [1988]. "Submodel selection and evaluation in regression: the x-fixed case and little bootstrap", Technical Report No. 169, Statistics Department, U.C. Berkeley.

Breiman, L. [1989]. "Additive models in regression using knot deletion, cross-validation and cost effectiveness" (in preparation).

Breiman, L. and Freedman, D. [1983]. "How many variables should be entered in a regression equation?" *JASA*, V.78, No. 381, 131-136.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. [1985]. "Classification and Regression Trees". Wadsworth.

Burman, A. [1989]. "Estimation of optimal transformations using v-fold cross-validation and repeated learning testing methods". To appear, *Sankhya, Ser. A.*

Efron, B. [1986]. "How biased is the apparent error rate of a prediction rule?" *JASA*, **81**, 461-470.

Miller, A.J. (1984). "Selection of subsets of regression variables" (with discussion). *J.R. Statist. Soc. A.*, **147**, part 2, 398-425.

Thompson, M.L. (1978). "Selection of variables in multiple regression". *International Statistical Review*, **46**, 1-49 and 129-146.

Wu, C.F.J. (1986). "Jackknife, bootstrap, and other resampling methods in regression analysis" (with discussion). *Ann. Statistics*, **14**, 1261-1350.

Figure 1

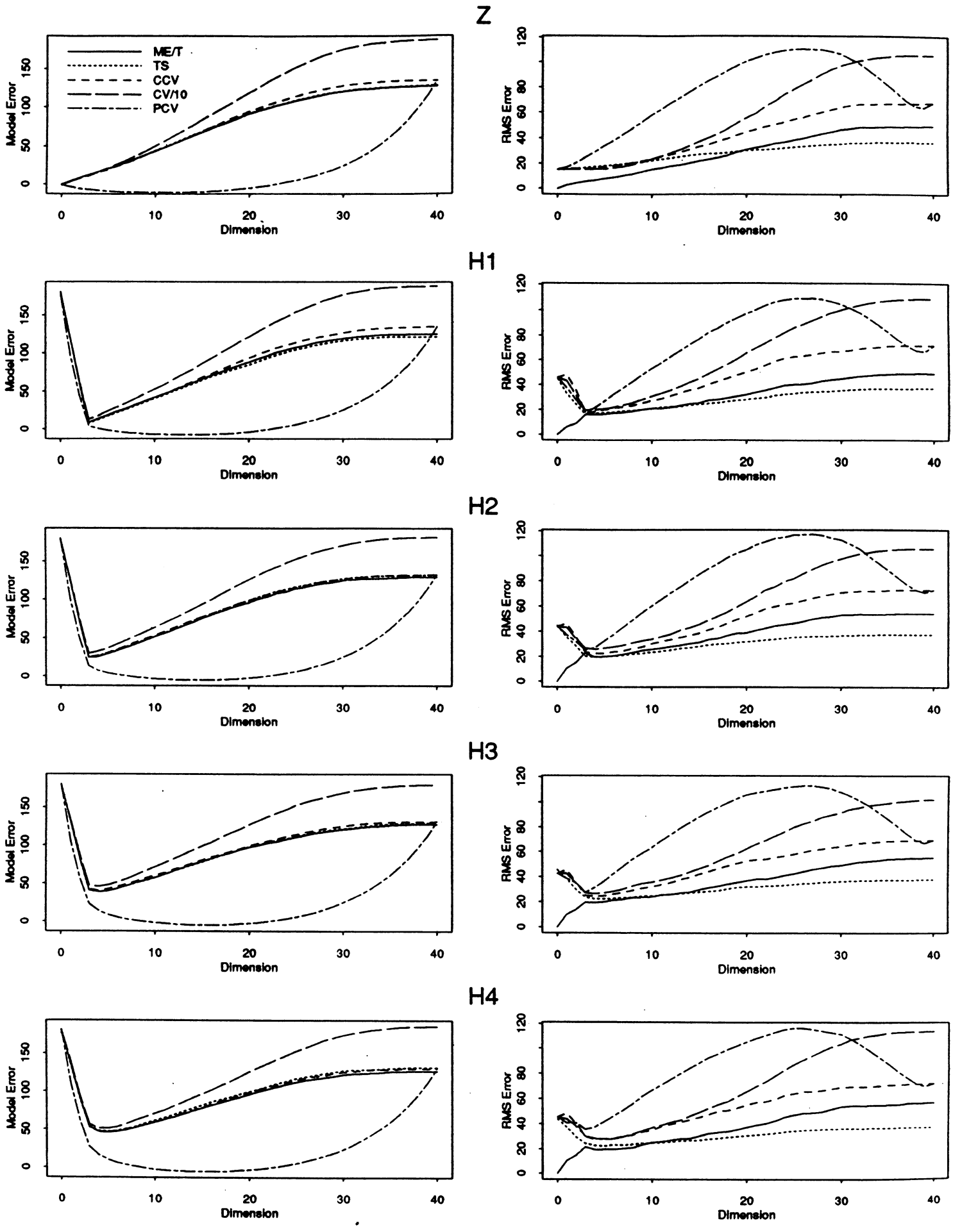

Normal N=60
Z
H1
H2
H3
H4
ME/T
TS
CCV
CV/10
PCV
Model Error
RMS Error
Dimension
0
10
20
30
40
0
50
100
150
0
20
40
60
80
120

Figure 2

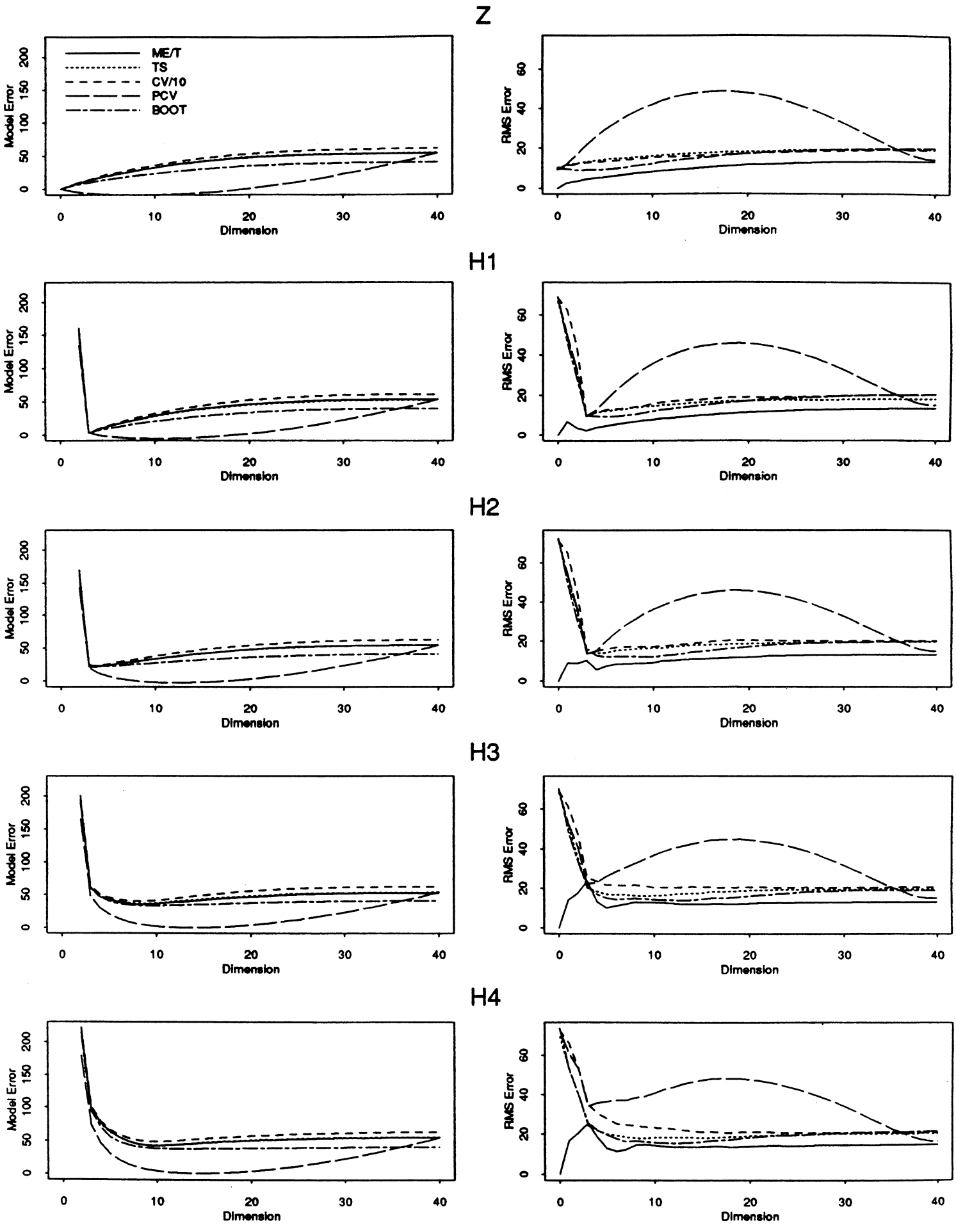

Normal N=160
Z
H1
H2
H3
H4
ME/T
TS
CV/10
PCV
BOOT
Model Error
RMS Error
Dimension
0
10
20
30
40
50
100
150
200
60

Figure 3

Lognormal N=60

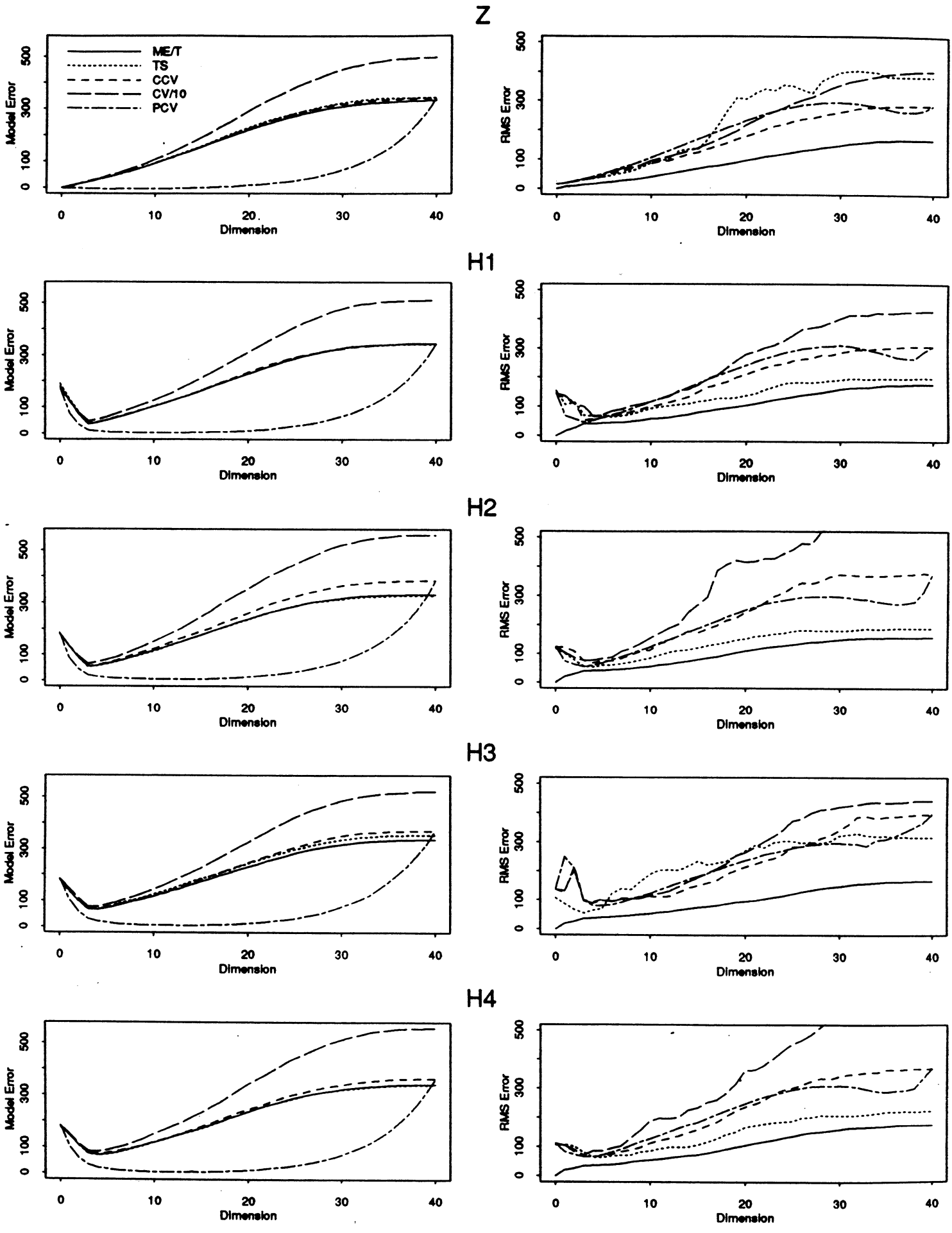


Figure 4

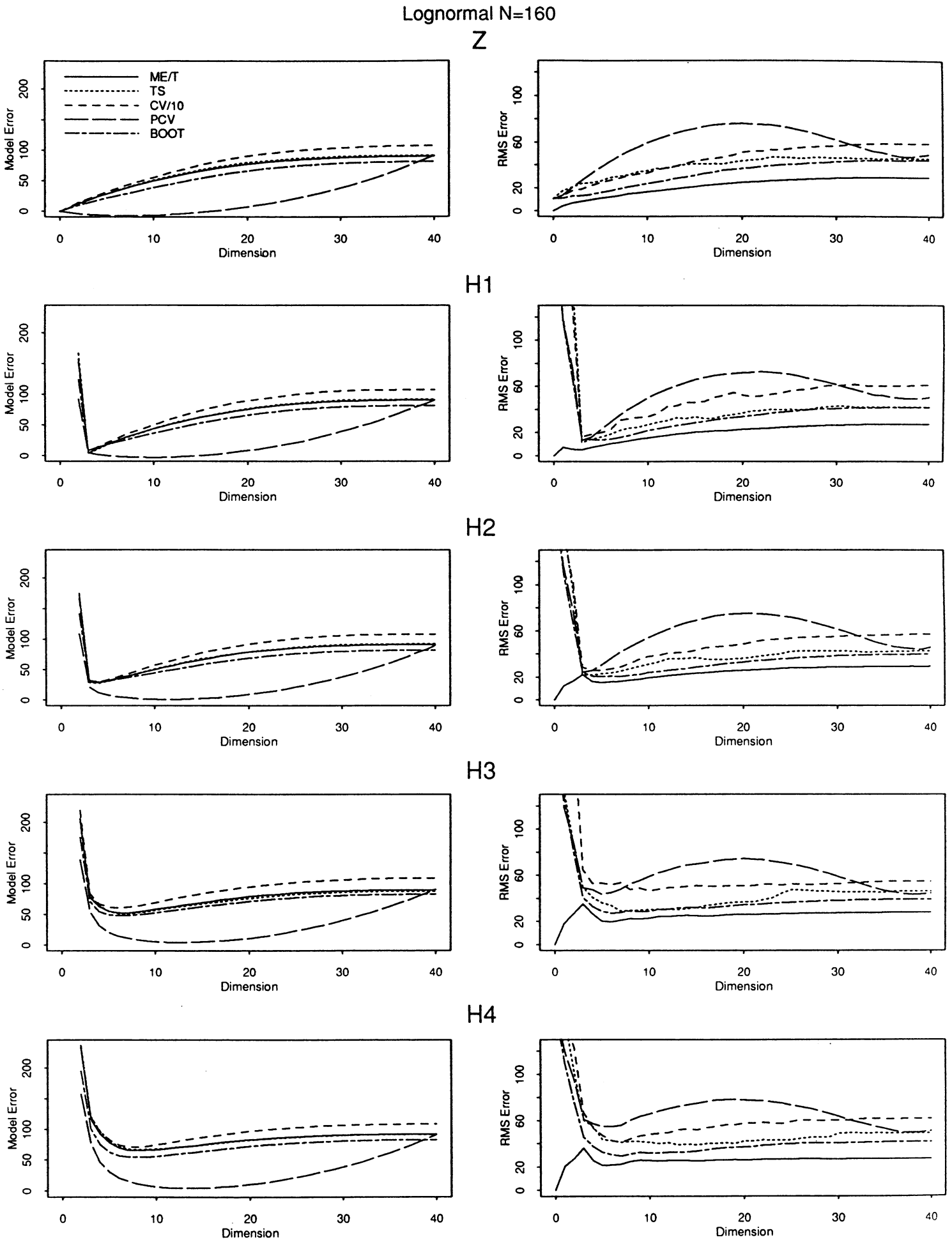
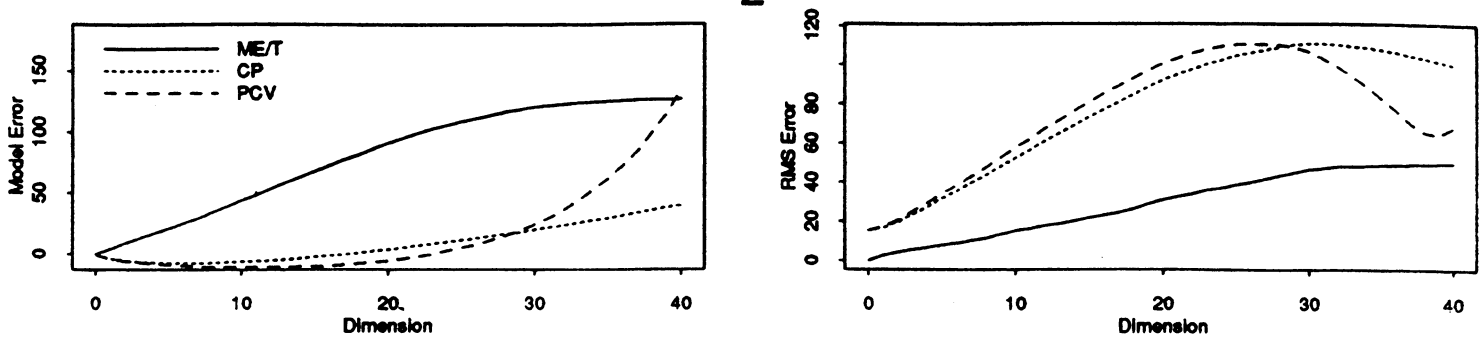

Lognormal N=160
Z
H1
H2
H3
H4
ME/T
TS
CV/10
PCV
BOOT
Model Error
RMS Error
Dimension
0
10
20
30
40
0
50
100
200
0
20
60
100

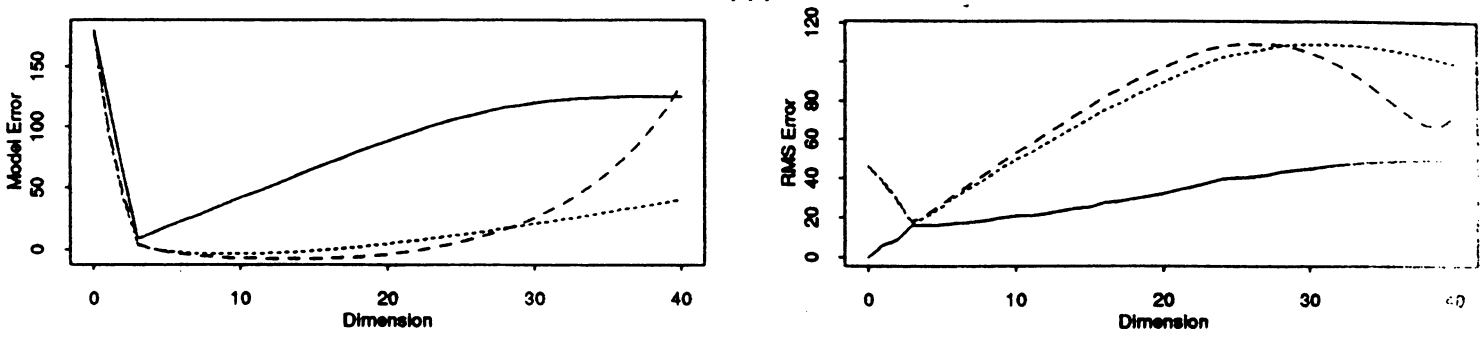
Figure 5

Normal N=60

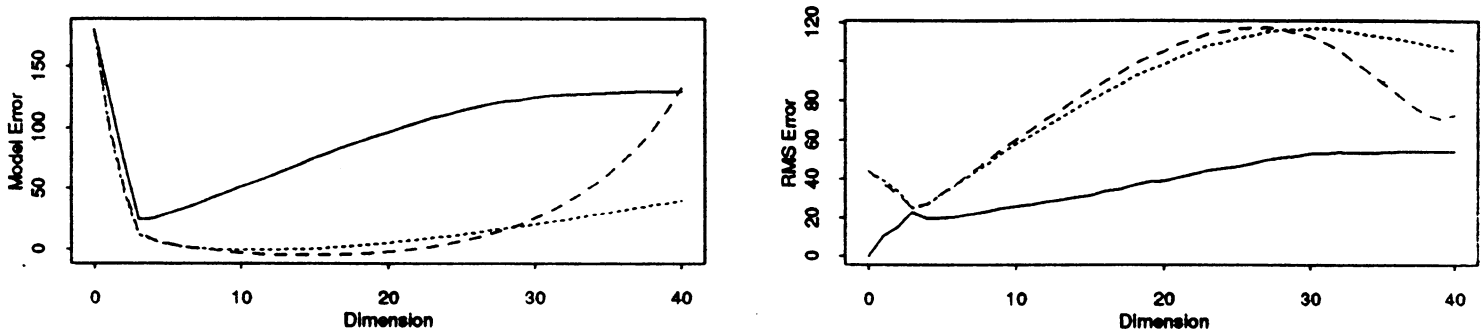
Z



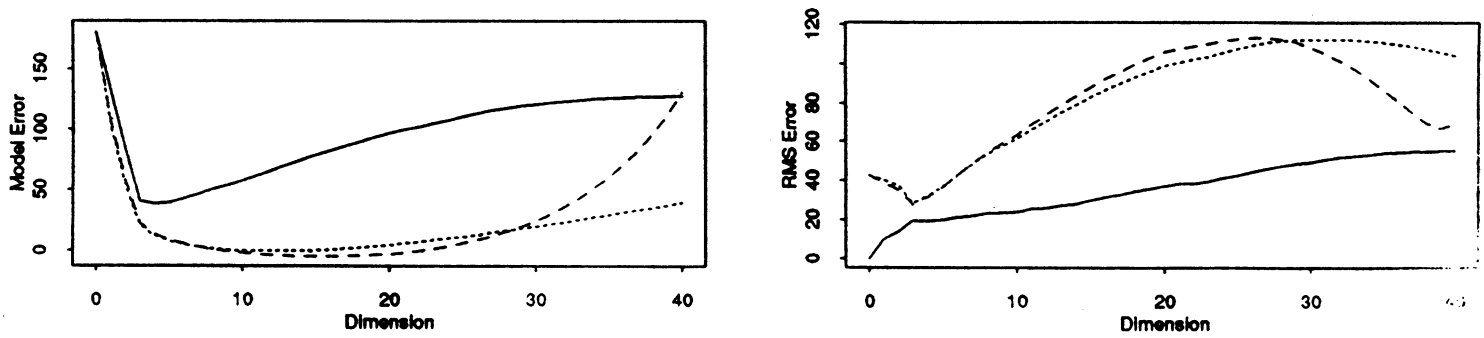
H1



H2



H3



H4

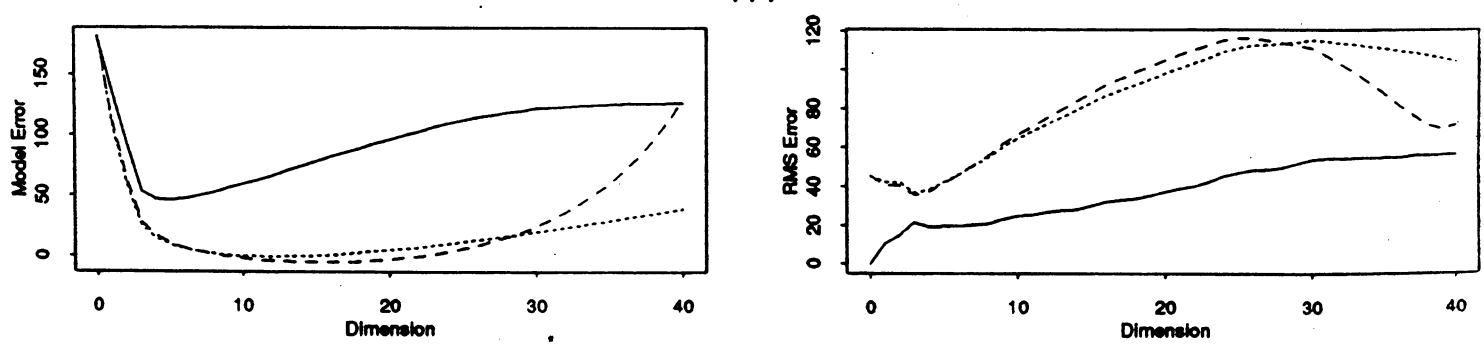


Figure 6

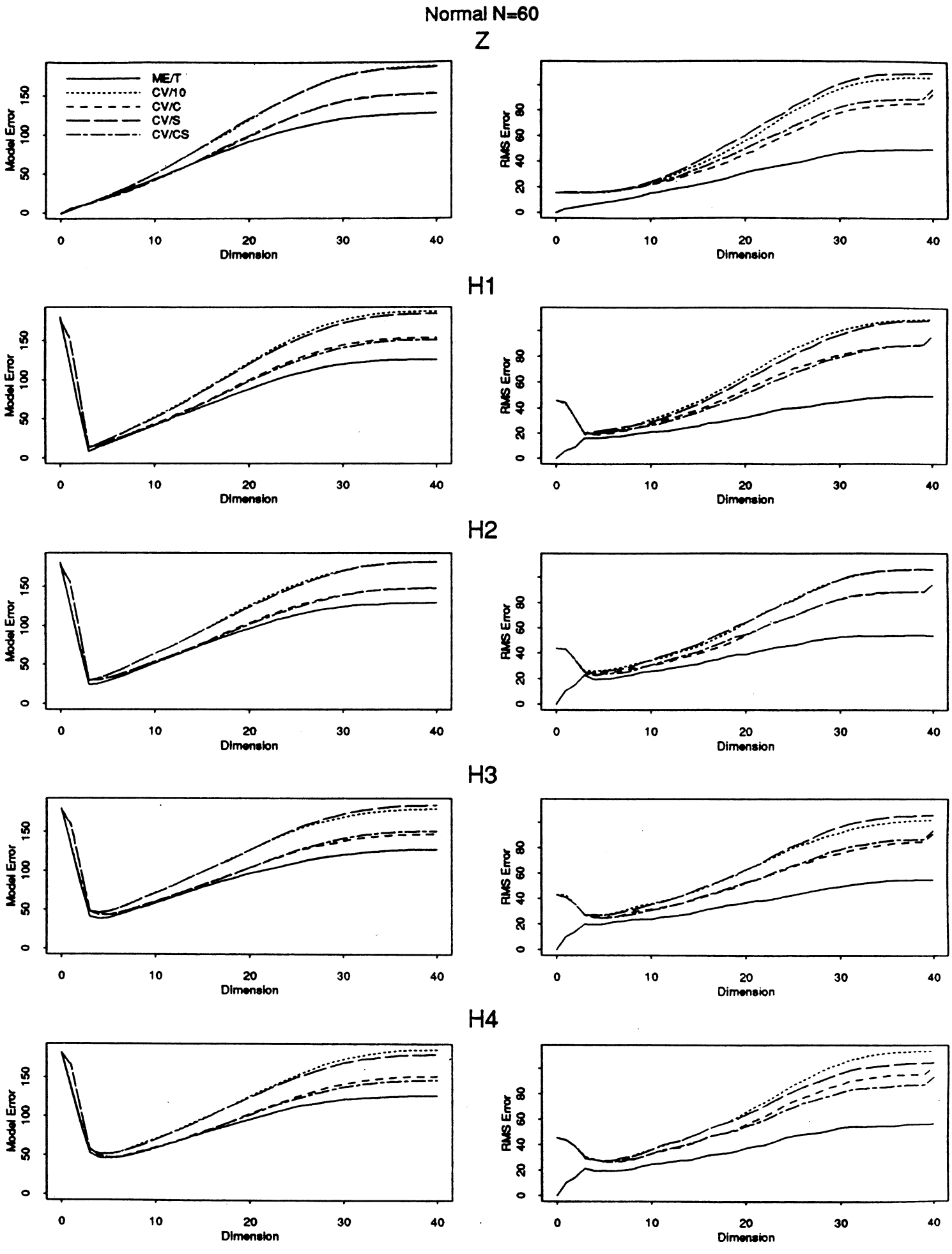
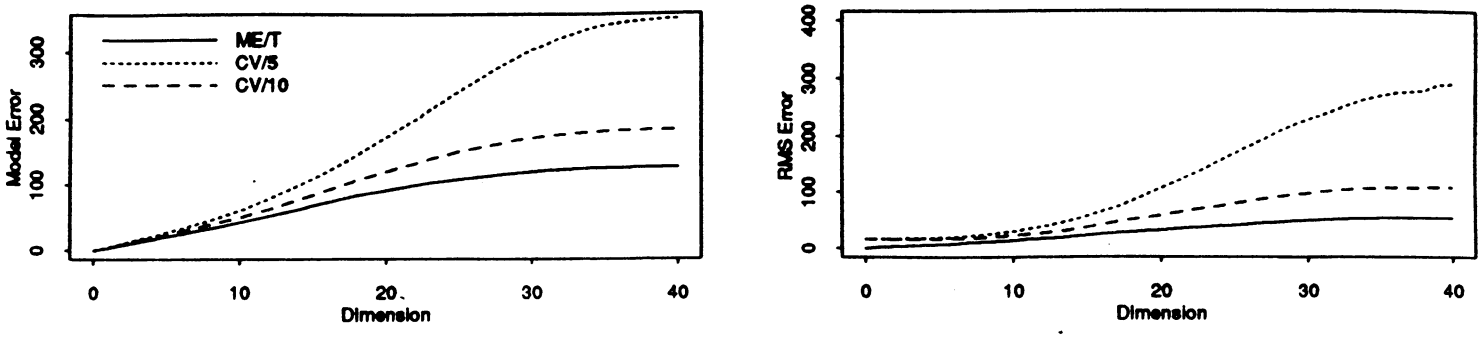

Normal N=60
Z
H1
H2
H3
H4
ME/T
CV/10
CV/C
CV/S
CV/CS
Model Error
RMS Error
Dimension
0
10
20
30
40
0
50
100
150
0
20
40
60
80

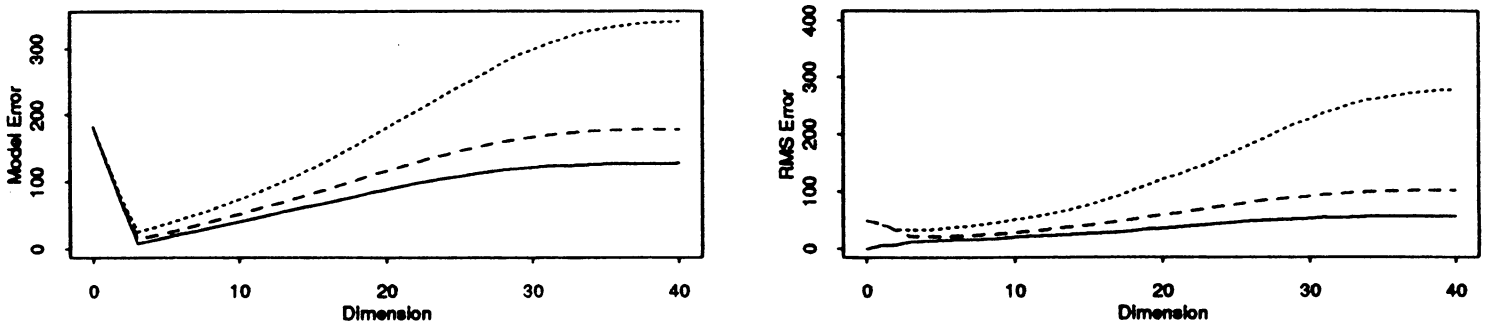
Figure 7

Normal N=60

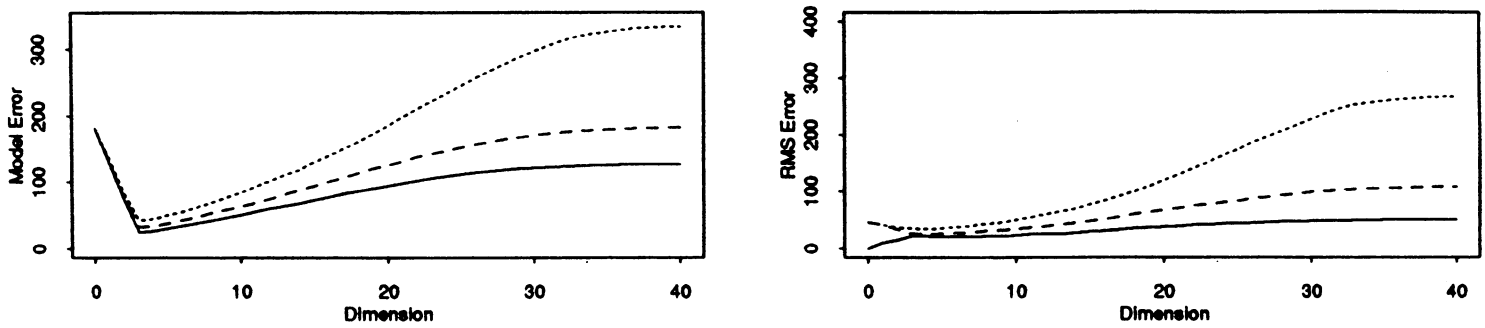
Z



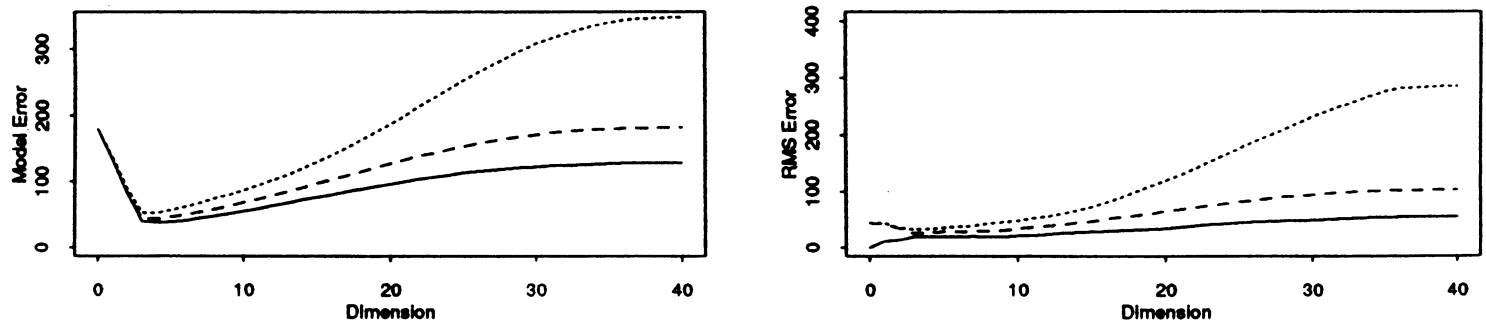
H1



H2



H3



H4

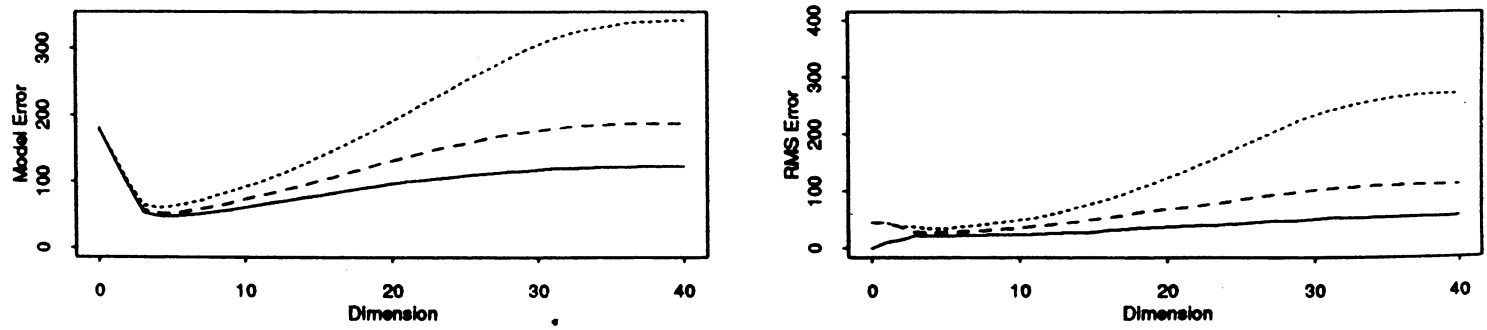


Figure 8

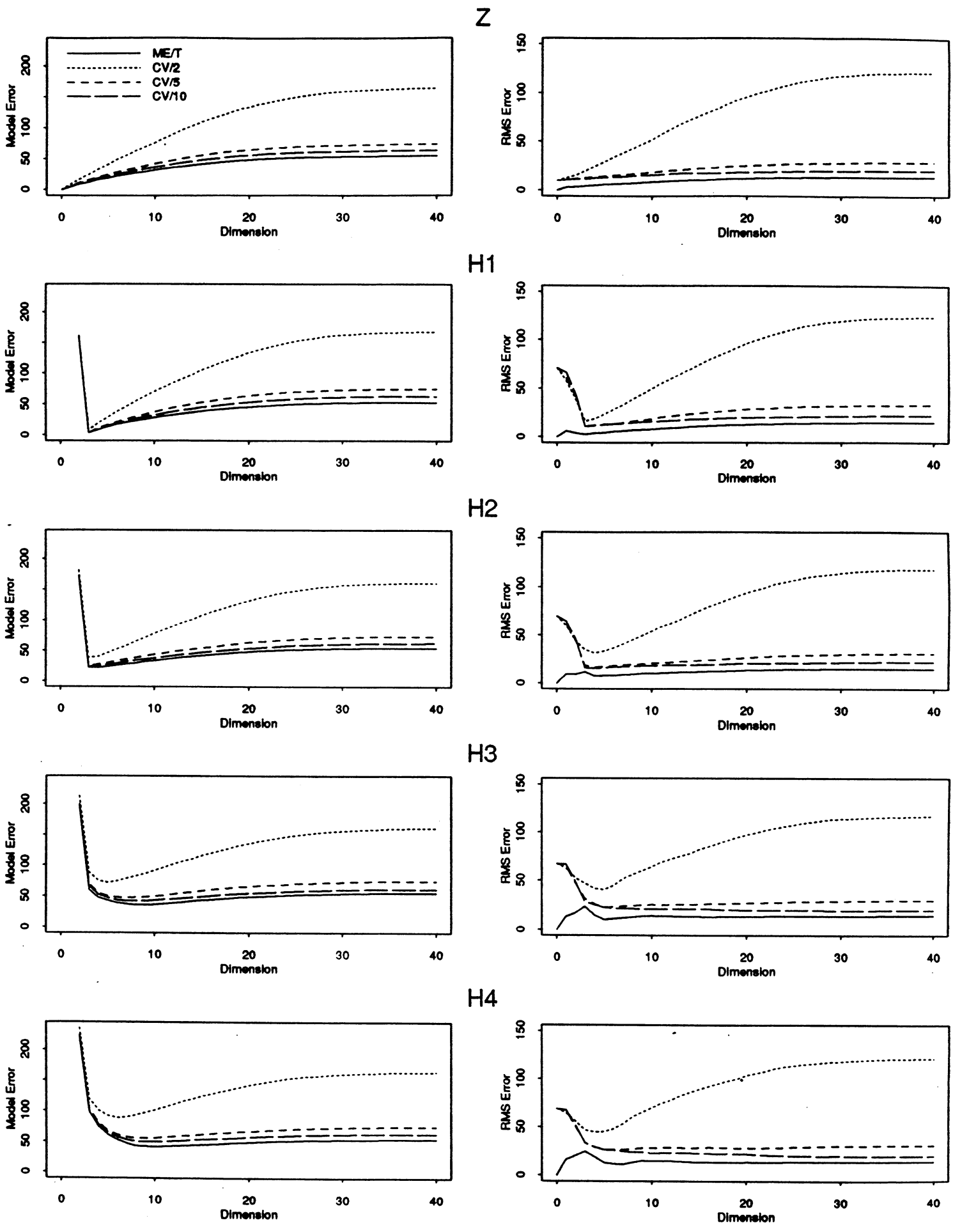
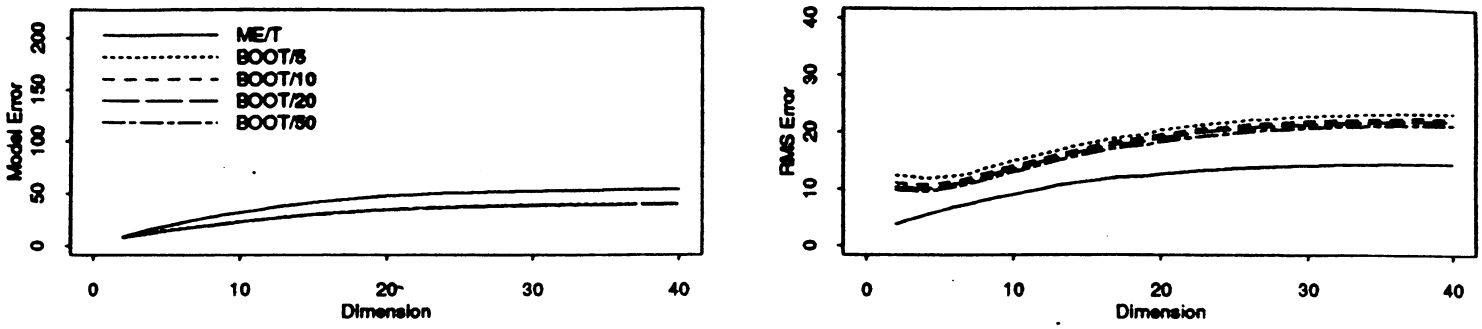

Normal N=160
Z
H1
H2
H3
H4
ME/T
CV/2
CV/5
CV/10
Model Error
RMS Error
Dimension
0
10
20
30
40
50
100
150
200

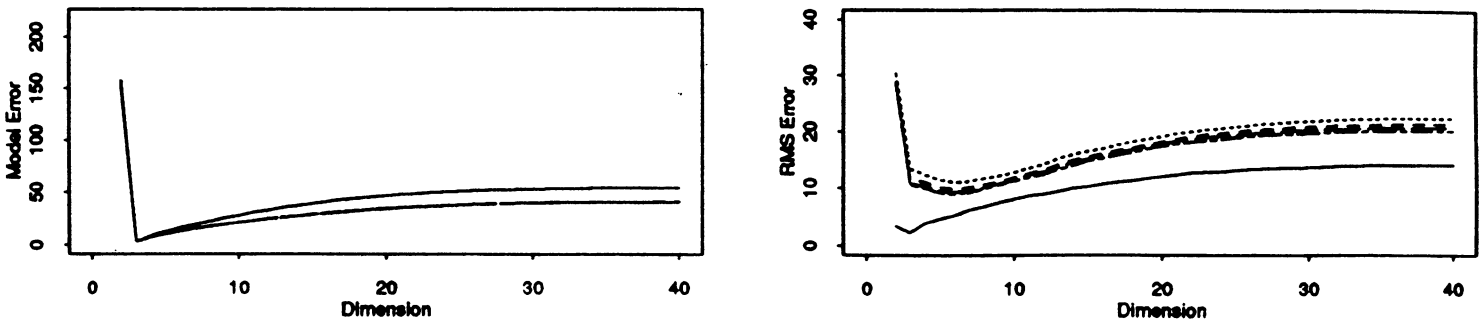
Figure 9

Normal N=160

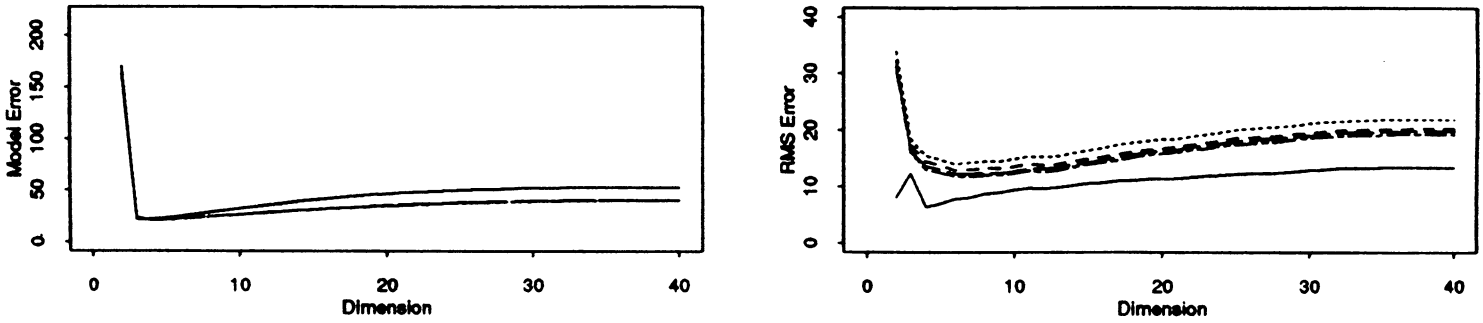
Z



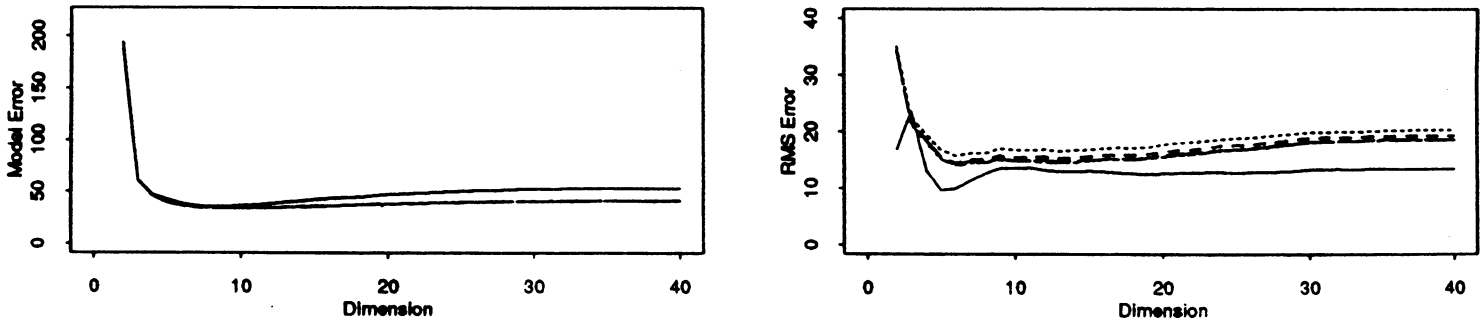
H1



H2



H3



H4

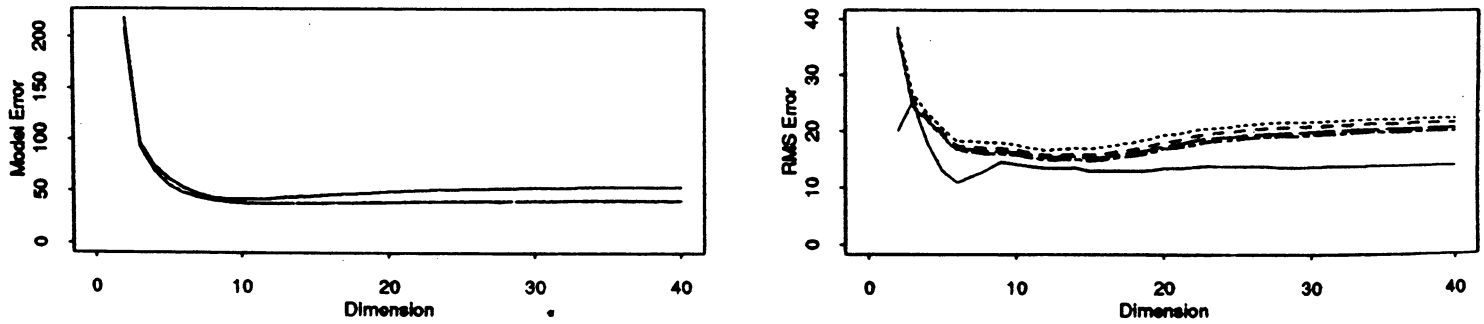
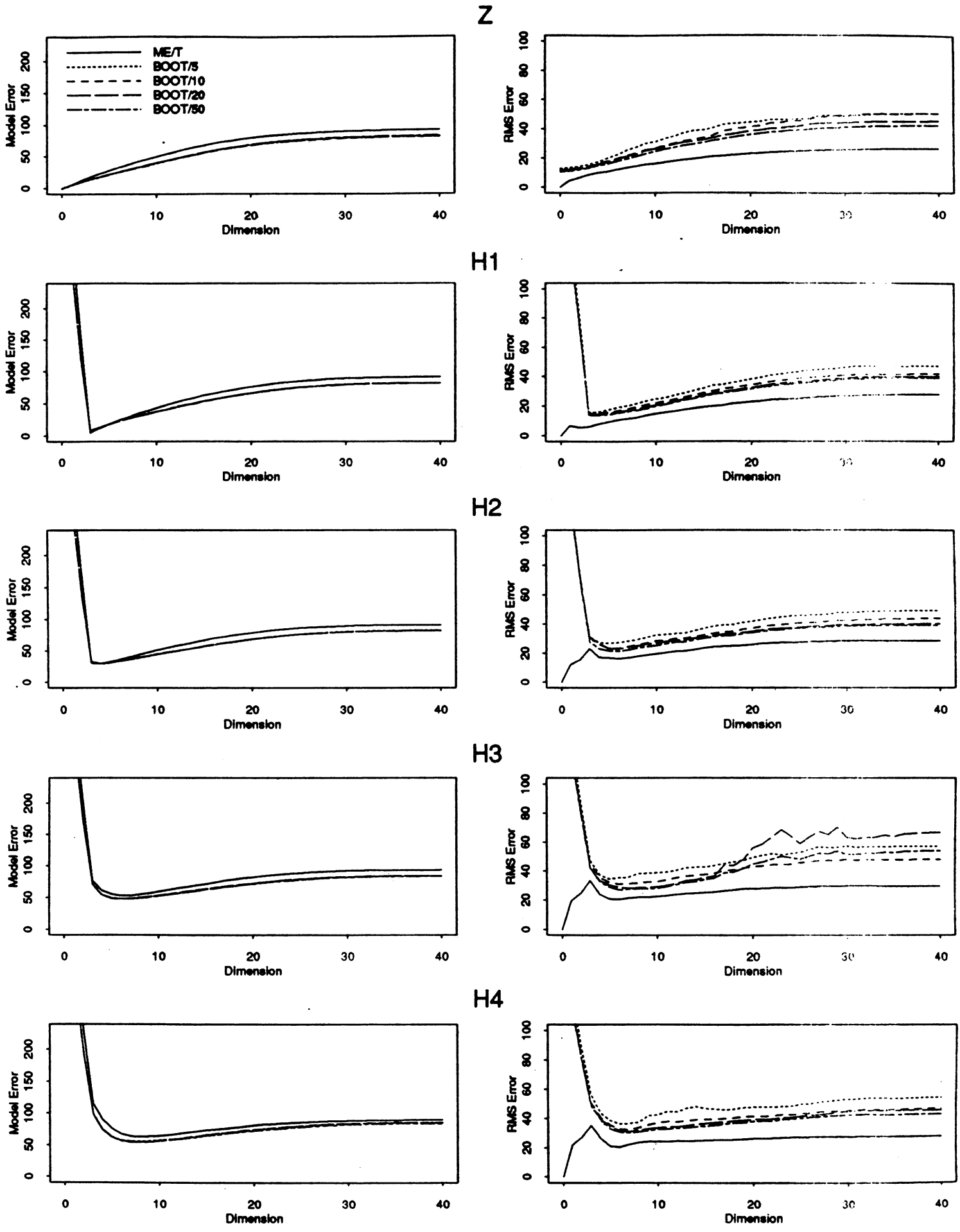


Figure 10


Lognormal N=160
Z
H1
H2
H3
H4
ME/T
BOOT/5
BOOT/10
BOOT/20
BOOT/50
Model Error
RMS Error
Dimension
0
10
20
30
40
0
50
100
150
200
0
20
40
60
80
100