Chapter 14
The ENTROPY Procedure (Experimental)

Contents

Overview: ENTROPY Procedure .............................................. 786
Getting Started: ENTROPY Procedure .................................. 788
  Simple Regression Analysis ............................................ 788
  Using Prior Information ................................................ 794
  Pure Inverse Problems .................................................. 799
  Analyzing Multinomial Response Data ................................. 804
Syntax: ENTROPY Procedure ............................................... 808
  Functional Summary ....................................................... 808
  PROC ENTROPY Statement ............................................. 810
  BOUNDS Statement ....................................................... 813
  BY Statement ............................................................. 815
  ID Statement .............................................................. 815
  MODEL Statement ......................................................... 815
  PRIORS Statement ........................................................ 816
  RESTRICT Statement ...................................................... 817
  TEST Statement ........................................................... 817
  WEIGHT Statement ........................................................ 819
Details: ENTROPY Procedure ............................................... 819
  Generalized Maximum Entropy ......................................... 819
  Generalized Cross Entropy .............................................. 820
  Moment Generalized Maximum Entropy ................................. 822
  Maximum Entropy-Based Seemingly Unrelated Regression .......... 823
  Generalized Maximum Entropy for Multinomial Discrete Choice Models ........................................ 825
  Censored or Truncated Dependent Variables ......................... 826
  Information Measures ..................................................... 827
  Parameter Covariance For GCE ......................................... 828
  Parameter Covariance For GCE-M ...................................... 828
  Statistical Tests ......................................................... 829
  Missing Values ............................................................ 829
  Input Data Sets ........................................................... 830
  Output Data Sets ......................................................... 831
  ODS Table Names ........................................................ 832
  ODS Graphics ............................................................. 832
Examples: ENTROPY Procedure ............................................. 833
  Example 14.1: Nonnormal Error Estimation .......................... 833
Overview: ENTROPY Procedure

The ENTROPY procedure implements a parametric method of linear estimation based on generalized maximum entropy. The ENTROPY procedure is suitable when there are outliers in the data and robustness is required, when the model is ill-posed or under-determined for the observed data, or for regressions that involve small data sets.

The main features of the ENTROPY procedure are as follows:

- estimation of simultaneous systems of linear regression models
- estimation of Markov models
- estimation of seemingly unrelated regression (SUR) models
- estimation of unordered multinomial discrete Choice models
- solution of pure inverse problems
- allowance of bounds and restrictions on parameters
- performance of tests on parameters
- allowance of data and moment constrained generalized cross entropy

It is often the case that the statistical/economic model of interest is ill-posed or under-determined for the observed data. For the general linear model, this can imply that high degrees of collinearity exist among explanatory variables or that there are more parameters to estimate than observations available to estimate them. These conditions lead to high variances or non-estimability for traditional generalized least squares (GLS) estimates.

Under these situations it might be in the researcher’s or practitioner’s best interest to consider a nontraditional technique for model fitting. The principle of maximum entropy is the foundation for an estimation methodology that is characterized by its robustness to ill-conditioned designs and its ability to fit over-parameterized models. See Mittelhammer, Judge, and Miller (2000) and Golan, Judge, and Miller (1996) for a discussion of Shannon’s maximum entropy measure and the related Kullback-Leibler information.

Generalized maximum entropy (GME) is a means of selecting among probability distributions to choose the distribution that maximizes uncertainty or uniformity remaining in the distribution, subject to information already known about the distribution. Information takes the form of data or moment constraints in the estimation procedure. PROC ENTROPY creates a GME distribution for each parameter in the linear model, based upon support points supplied by the user. The mean of each distribution is used as the estimate of the
parameter. Estimates tend to be biased, as they are a type of shrinkage estimate, but typically portray smaller variances than ordinary least squares (OLS) counterparts, making them more desirable from a mean squared error viewpoint (see Figure 14.1).

**Figure 14.1** Distribution of Maximum Entropy Estimates versus OLS

Maximum entropy techniques are most widely used in the econometric and time series fields. Some important uses of maximum entropy include the following:

- size distribution of firms
- stationary Markov Process
- social accounting matrix (SAM)
- consumer brand preference
- exchange rate regimes
- wage-dependent firm relocation
- oil market dynamics
Getting Started: ENTROPY Procedure

This section introduces the ENTROPY procedure and shows how to use PROC ENTROPY for several kinds of statistical analyses.

Simple Regression Analysis

The ENTROPY procedure is similar in syntax to the other regression procedures in SAS. To demonstrate the similarity, suppose the endogenous/dependent variable is $y$, and $x_1$ and $x_2$ are two exogenous/independent variables of interest. To estimate the parameters in this single equation model using PROC ENTROPY, use the following SAS statements:

```sas
proc entropy;
   model y = x1 x2;
run;
```

Test Scores Data Set

Consider the following test score data compiled by Coleman et al. (1966):

```sas
title "Test Scores compiled by Coleman et al. (1966)";
data coleman;
   input test_score 6.2 teach_sal 6.2 prcnt_prof 8.2 socio_stat 9.2 teach_score 8.2 mom_ed 7.2;
   label test_score="Average sixth grade test scores in observed district";
   label teach_sal="Average teacher salaries per student (1000s of dollars)"
      prcnt_prof="Percent of students' fathers with professional employment"
      socio_stat="Composite measure of socio-economic status in the district"
      teach_score="Average verbal score for teachers"
      mom_ed="Average level of education (years) of the students' mothers"
   datalines;
   37.01 3.83 28.87 7.20 26.60 6.19 ... more lines ...
```

This data set contains outliers, and the condition number of the matrix of regressors, $X$, is large, which indicates collinearity among the regressors. Since the maximum entropy estimates are both robust with respect to the outliers and also less sensitive to a high condition number of the $X$ matrix, maximum entropy estimation is a good choice for this problem.

To fit a simple linear model to this data by using PROC ENTROPY, use the following statements:

```sas
proc entropy data=coleman;
   model test_score = teach_sal prcnt_prof socio_stat teach_score mom_ed;
run;
```
This requests the estimation of a linear model for TEST_SCORE with the following form:

\[
\text{test\_score} = \text{intercept} + a \cdot \text{teach\_sal} + b \cdot \text{prcnt\_prof} + c \cdot \text{socio\_stat} \\
+ d \cdot \text{teach\_score} + e \cdot \text{mom\_ed} + \epsilon;
\]

This estimation produces the “Model Summary” table in Figure 14.2, which shows the equation variables used in the estimation.

**Figure 14.2** Model Summary Table

**Test Scores compiled by Coleman et al. (1966)**

**The ENTROPY Procedure**

<table>
<thead>
<tr>
<th>Variables(Supports(Weights))</th>
<th>teach_sal</th>
<th>prcnt_prof</th>
<th>socio_stat</th>
<th>teach_score</th>
<th>mom_ed</th>
<th>Intercept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equations(Supports(Weights))</td>
<td>test_score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Since support points and prior weights are not specified in this example, they are not shown in the “Model Summary” table. The next four pieces of information displayed in Figure 14.3 are: the “Data Set Options,” the “Minimization Summary,” the “Final Information Measures,” and the “Observations Processed.”

**Figure 14.3** Estimation Summary Tables

**Test Scores compiled by Coleman et al. (1966)**

**The ENTROPY Procedure**

**GME Estimation Summary**

<table>
<thead>
<tr>
<th>Data Set Options</th>
<th>DATA= WORK.COLEMAN</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Minimization Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters Estimated</td>
</tr>
<tr>
<td>Covariance Estimator</td>
</tr>
<tr>
<td>Entropy Type</td>
</tr>
<tr>
<td>Entropy Form</td>
</tr>
<tr>
<td>Numerical Optimizer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Final Information Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective Function Value</td>
</tr>
<tr>
<td>Signal Entropy</td>
</tr>
<tr>
<td>Noise Entropy</td>
</tr>
<tr>
<td>Normed Entropy (Signal)</td>
</tr>
<tr>
<td>Normed Entropy (Noise)</td>
</tr>
<tr>
<td>Parameter Information Index</td>
</tr>
<tr>
<td>Error Information Index</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Observations Processed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
</tr>
<tr>
<td>Used</td>
</tr>
</tbody>
</table>
The item labeled “Objective Function Value” is the value of the entropy estimation criterion for this estimation problem. This measure is analogous to the log-likelihood value in a maximum likelihood estimation. The “Parameter Information Index” and the “Error Information Index” are normalized entropy values that measure the proximity of the solution to the prior or target distributions.

The next table displayed is the ANOVA table, shown in Figure 14.4. This is in the same form as the ANOVA table for the MODEL procedure, since this is also a multivariate procedure.

**Figure 14.4** Summary of Residual Errors

<table>
<thead>
<tr>
<th>Equation</th>
<th>DF Model</th>
<th>Error</th>
<th>SSE</th>
<th>MSE</th>
<th>Root MSE</th>
<th>R-Square</th>
<th>Adj RSq</th>
</tr>
</thead>
<tbody>
<tr>
<td>test_score</td>
<td>6</td>
<td>14</td>
<td>175.8</td>
<td>8.7881</td>
<td>2.9645</td>
<td>0.7266</td>
<td>0.6290</td>
</tr>
</tbody>
</table>

The last table displayed is the “Parameter Estimates” table, shown in Figure 14.5. The difference between this parameter estimates table and the parameter estimates table produced by other regression procedures is that the standard error and the probabilities are labeled as approximate.

**Figure 14.5** Parameter Estimates

| Variable | Approx Estimate | Approx Std Err | Approx t Value | Approx Pr > |t| |
|----------|-----------------|----------------|----------------|-------------|-----|
| teach_sal | 0.287979 | 0.00551 | 52.26 | <.0001 |
| prcnt_prof | 0.02266 | 0.00323 | 7.01 | <.0001 |
| socio_stat | 0.199777 | 0.0308 | 6.48 | <.0001 |
| teach_score | 0.497137 | 0.0180 | 27.61 | <.0001 |
| mom_ed | 1.644472 | 0.0921 | 17.85 | <.0001 |
| Intercept | 10.5021 | 0.3958 | 26.53 | <.0001 |
The parameter estimates produced by the REG procedure for this same model are shown in Figure 14.6. Note that the parameters and standard errors from PROC REG are much different than estimates produced by PROC ENTROPY.

```
symbol v=dot h=1 c=green;

proc reg data=coleman;
  model test_score = teach_sal prcnt_prof socio_stat teach_score mom_ed;
  plot rstudent.*obs. / vref= -1.714 1.714 cvref=blue lvref=1
    HREF=0 to 30 by 5 cHREF=red cframe=ligr;
run;
```

**Figure 14.6** REG Procedure Parameter Estimates

**Test Scores compiled by Coleman et al. (1966)**

<table>
<thead>
<tr>
<th>Parameter Estimates</th>
<th>Parameter Estimates</th>
<th>Parameter Estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>DF</td>
<td>Estimate</td>
</tr>
<tr>
<td>Intercept</td>
<td>1</td>
<td>19.94857</td>
</tr>
<tr>
<td>teach_sal</td>
<td>1</td>
<td>1.79333</td>
</tr>
<tr>
<td>prcnt_prof</td>
<td>1</td>
<td>0.04360</td>
</tr>
<tr>
<td>socio_stat</td>
<td>1</td>
<td>0.55576</td>
</tr>
<tr>
<td>teach_score</td>
<td>1</td>
<td>1.11017</td>
</tr>
<tr>
<td>mom_ed</td>
<td>1</td>
<td>-1.81092</td>
</tr>
</tbody>
</table>

This data set contains two outliers, observations 3 and 18. These can be seen in a plot of the residuals shown in Figure 14.7
The presence of outliers suggests that a robust estimator such as $M$-estimator in the ROBUSTREG procedure should be used. The following statements use the ROBUSTREG procedure to estimate the model.

```plaintext
proc robustreg data=coleman;
   model test_score = teach_sal prcnt_prof
                     socio_stat teach_score mom_ed;
run;
```

The results of the estimation are shown in Figure 14.8.
Note that TEACH_SAL(VAR1) and MOM_ED(VAR5) change greatly when the robust estimation is used. Unfortunately, these two coefficients are negative, which implies that the test scores increase with decreasing teacher salaries and decreasing levels of the mother’s education. Since ROBUSTREG is robust to outliers, they are not causing the counterintuitive parameter estimates.

The condition number of the regressor matrix $X$ also plays an important role in parameter estimation. The condition number of the matrix can be obtained by specifying the COLLIN option in the PROC ENTROPY statement.

```r
proc entropy data=coleman collin;
    model test_score = teach_sal prcnt_prof socio_stat teach_score mom_ed;
run;
```

The output produced by the COLLIN option is shown in Figure 14.9.

The condition number of the $X$ matrix is reported to be 84.85. This means that the condition number of $X'X$ is $84.85^2 = 7199.5$, which is very large.
Ridge regression can be used to offset some of the problems associated with ill-conditioned \( X \) matrices. Using the formula for the ridge value as
\[
\lambda_R = \frac{kS^2}{\hat{\beta}'\hat{\beta}} \approx 0.9
\]

where \( \hat{\beta} \) and \( S^2 \) are the least squares estimators of \( \beta \) and \( \sigma^2 \) and \( k = 6 \). A ridge regression of the test score model was performed by using the data set with the outliers removed. The following PROC REG code performs the ridge regression:

```plaintext
data coleman;
set coleman;
  if _n_ = 3 or _n_ = 18 then delete;
run;

proc reg data=coleman ridge=0.9 outest=t noprint;
  model test_score = teach_sal prcnt_prof socio_stat teach_score mom_ed;
run;

proc print data=t;
run;
```

The results of the estimation are shown in Figure 14.10.

**Figure 14.10** Ridge Regression Estimates

<table>
<thead>
<tr>
<th>Obs</th>
<th><em>MODEL</em></th>
<th><em>TYPE</em></th>
<th><em>DEPVAR</em></th>
<th><em>RIDGE</em></th>
<th><em>PCOMIT</em></th>
<th><em>RMSE</em></th>
<th>Intercept</th>
<th>teach_sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MODEL1</td>
<td>PARMS</td>
<td>test_score</td>
<td>.</td>
<td>.</td>
<td>0.78236</td>
<td>29.7577</td>
<td>-1.69854</td>
</tr>
<tr>
<td>2</td>
<td>MODEL1</td>
<td>RIDGE</td>
<td>test_score</td>
<td>0.9</td>
<td></td>
<td>3.19679</td>
<td>9.6698</td>
<td>-0.08892</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>prcnt_prof</th>
<th>socio_stat</th>
<th>teach_score</th>
<th>mom_ed</th>
<th>test_score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.085118</td>
<td>0.66617</td>
<td>1.18400</td>
<td>-4.06675</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>0.041889</td>
<td>0.23223</td>
<td>0.60041</td>
<td>1.32168</td>
<td>-1</td>
</tr>
</tbody>
</table>

Note that the ridge regression estimates are much closer to the estimates produced by the ENTROPY procedure that uses the original data set. Ridge regressions are not robust to outliers as maximum entropy estimates are. This might explain why the estimates still differ for TEACH_SAL.

**Using Prior Information**

You can use prior information about the parameters or the residuals to improve the efficiency of the estimates. Some authors prefer the terms *pre-sample* or *pre-data* over the term *prior* when used with maximum entropy to avoid confusion with Bayesian methods. The maximum entropy method described here does not use Bayes’ rule when including prior information in the estimation.

To perform regression, the ENTROPY procedure uses a generalization of maximum entropy called *generalized maximum entropy*. In maximum entropy estimation, the unknowns are probabilities. Generalized maximum entropy expands the set of problems that can be solved by introducing the concept of *support points*. 
Generalized maximum entropy still estimates probabilities, but these are the probabilities of a support point. Support points are used to map the (0, 1) domain of the maximum entropy to the any finite range of values.

Prior information, such as expected ranges for the parameters or the residuals, is added by specifying support points for the parameters or the residuals. Support points are points in one dimension that specify the expected domain of the parameter or the residual. The wider the domain specified, the less efficient your parameter estimates are (the more variance they have). Specifying more support points in the same width interval also improves the efficiency of the parameter estimates at the cost of more computation. Golan, Judge, and Miller (1996) show that the gains in efficiency fall off for adding more than five support points. You can specify between 2 to 256 support points in the ENTROPY procedure.

If you have only a small amount of data, the estimates are very sensitive to your selection of support points and weights. For larger data sets, incorrect priors are discounted if they are not supported by the data.

Consider the data set generated by the following SAS statements:

```sas
data prior;
  do by = 1 to 100;
    do t = 1 to 10;
      y = 2*t + 5 * rannor(4);
      output;
    end;
  end;
run;
```

The PRIOR data set contains 100 samples of 10 observations each from the population

\[ y = 2 \cdot t + \epsilon \]
\[ \epsilon \sim N(0, 5) \]

You can estimate these samples using PROC ENTROPY as

```sas
proc entropy data=prior outest=parml noprint;
  model y = t ;
  by by;
run;
```

The 100 estimates are summarized by using the following SAS statements:

```sas
proc univariate data=parml;
  var t;
run;
```

The summary statistics from PROC UNIVARIATE are shown in Output 14.11. The true value of the coefficient \( T \) is 2.0, demonstrating that maximum entropy estimates tend to be biased.
Now assume that you have prior information about the slope and the intercept for this model. You are reasonably confident that the slope is 2 and you are less confident that intercept is zero. To specify prior information about the parameters, use the PRIORS statement.

There are two parts to the prior information specified in the PRIORS statement. The first part is the support points for a parameter. The support points specify the domain of the parameter. For example, the following statement sets the support points -1000 and 1000 for the parameter associated with variable T:

```
priors t -1000 1000;
```

This means that the coefficient lies in the interval $[-1000, 1000]$. If the estimated value of the coefficient is actually outside of this interval, the estimation will not converge. In the previous PRIORS statement, no weights were specified for the support points, so uniform weights are assumed. This implies that the coefficient has a uniform probability of being in the interval $[-1000, 1000]$.

The second part of the prior information is the weights on the support points. For example, the following statements sets the support points 10, 15, 20, and 25 with weights 1, 5, 5, and 1 respectively for the coefficient of T:

```
priors t 10(1) 15(5) 20(5) 25(1);
```

This creates the prior distribution on the coefficient shown in Figure 14.12. The weights are automatically normalized so that they sum to one.
For the PRIOR data set created previously, the expected value of the coefficient of \( T \) is 2. The following SAS statements reestimate the parameters with a prior weight specified for each one.

```
proc entropy data=prior outest=parm2 noprint;
   priors t 0(1) 2(3) 4(1)
       intercept -100(.5) -10(1.5) 0(2) 10(1.5) 100(0.5);
   model y = t;
   by by;
run;
```

The priors on the coefficient of \( T \) express a confident view of the value of the coefficient. The priors on \textsc{intercept} express a more diffuse view on the value of the intercept. The following \textsc{PROC UNIVARIATE} statement computes summary statistics from the estimations:

```
proc univariate data=parm2;
   var t;
run;
```

The summary statistics for the distribution of the estimates of \( T \) are shown in Figure 14.13.
The prior information improves the estimation of the coefficient of $T$ dramatically. The downside of specifying priors comes when they are incorrect. For example, say the priors for this model were specified as

```plaintext
priors t -2(1) 0(3) 2(1);
```

to indicate a prior centered on zero instead of two.

The resulting summary statistics shown in Figure 14.14 indicate how the estimation is biased away from the solution.
The more data available for estimation, the less sensitive the parameters are to the priors. If the number of observations in each sample is 50 instead of 10, then the summary statistics shown in Figure 14.15 are produced. The prior information is not supported by the data, so it is discounted.

![Figure 14.15 Incorrect Prior Information with More Data](image)

**Prior Distribution of Parameter T**

**The UNIVARIATE Procedure**

<table>
<thead>
<tr>
<th>Variable:</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Basic Statistical Measures</strong></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>Variability</td>
</tr>
<tr>
<td>Mean</td>
<td>0.652921</td>
</tr>
<tr>
<td>Median</td>
<td>0.653486</td>
</tr>
<tr>
<td>Mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Pure Inverse Problems**

A special case of systems of equations estimation is the pure inverse problem. A pure problem is one that contains an exact relationship between the dependent variable and the independent variables and does not have an error component. A pure inverse problem can be written as

\[ y = X\beta \]

where \( y \) is a \( n \)-dimensional vector of observations, \( X \) is a \( n \times k \) matrix of regressors, and \( \beta \) is a \( k \)-dimensional vector of unknowns. Notice that there is no error term.

A classic example is a dice problem (Jaynes 1963). Given a six-sided die that can take on the values \( x = 1, 2, 3, 4, 5, 6 \) and the average outcome of the die \( y = A \), compute the probabilities \( \beta = (p_1, p_2, \ldots, p_6)' \) of rolling each number. This infers six values from two pieces of information. The data points are the expected value of \( y \), and the sum of the probabilities is one. Given \( E(y) = 4.0 \), this problem is solved by using the following SAS code:
data one;
    array x[6] ( 1 2 3 4 5 6 );
y=4.0;
run;

proc entropy data=one pure;
priors x1 0 1 x2 0 1 x3 0 1 x4 0 1 x5 0 1 x6 0 1;
model y = x1-x6/ noint;
restrict x1 + x2 +x3 +x4 + x5 + x6 =1;
run;
The probabilities are given in Figure 14.16.

**Figure 14.16** Jaynes' Dice Pure Inverse Problem

*Prior Distribution of Parameter T*

<table>
<thead>
<tr>
<th>Variable</th>
<th>Estimate</th>
<th>Standard Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0.101763</td>
<td>0.5254</td>
</tr>
<tr>
<td>x2</td>
<td>0.122658</td>
<td>0.4630</td>
</tr>
<tr>
<td>x3</td>
<td>0.147141</td>
<td>0.3974</td>
</tr>
<tr>
<td>x4</td>
<td>0.175533</td>
<td>0.3298</td>
</tr>
<tr>
<td>x5</td>
<td>0.208066</td>
<td>0.2622</td>
</tr>
<tr>
<td>x6</td>
<td>0.244839</td>
<td>0.1970</td>
</tr>
</tbody>
</table>

Note how the probabilities are skewed to the higher values because of the high average roll provided in the input data.

**First-Order Markov Process Estimation**

A more useful inverse problem is the first-order markov process. Companies have a share of the marketplace where they do business. Generally, customers for a specific market space can move from company to company. The movement of customers can be visualized graphically as a flow diagram, as in Figure 14.17. The arrows represent movements of customers from one company to another.
You can model the probability that a customer moves from one company to another using a first-order Markov model. Mathematically the model is:

$$y_t = P y_{t-1}$$

where $y_t$ is a vector of $k$ market shares at time $t$ and $P$ is a $k \times k$ matrix of unknown transition probabilities. The value $p_{ij}$ represents the probability that a customer who is currently using company $j$ at time $t - 1$ moves to company $i$ at time $t$. The diagonal elements then represent the probability that a customer stays with the current company. The columns in $P$ sum to one.

Given market share information over time, you can estimate the transition probabilities $P$. In order to estimate $P$ using traditional methods, you need at least $k$ observations. If you have fewer than $k$ transitions, you can use the ENTROPY procedure to estimate the probabilities.

Suppose you are studying the market share for four companies. If you want to estimate the transition probabilities for these four companies, you need a time series with four observations of the shares. Assume the current transition probability matrix is as follows:

$$\begin{bmatrix}
0.7 & 0.4 & 0.0 & 0.1 \\
0.1 & 0.5 & 0.4 & 0.0 \\
0.0 & 0.1 & 0.6 & 0.0 \\
0.2 & 0.0 & 0.0 & 0.9
\end{bmatrix}$$

The following SAS DATA step statements generate a series of market shares from this probability matrix. A transition is represented as the current period shares, $y$, and the previous period shares, $x$. 

![Markov Transition Diagram]
data m;
    /* Known Transition matrix */
    array p[4,4] (0.7 .4 .0 .1  
    0.1 .5 .4 .0  
    0.0 .1 .6 .0  
    0.2 .0 .0 .9 );
    /* Initial Market shares */
    array y[4] y1-y4 ( .4 .3 .2 .1 );
    array x[4] x1-x4;
    drop p1-p16 i;
    do i = 1 to 3;
        x[1] = y[1]; x[2] = y[2];
        y[1] = p[1,1] * x1 + p[1,2] * x2 + p[1,3] * x3 + p[1,4] * x4;
        output;
    end;
run;

The following SAS statements estimate the transition matrix by using only the first transition.

    proc entropy markov pure data=m(obs=1);
        model y1-y4 = x1-x4;
    run;

The MARKOV option implies NOINT for each model, that the sum of the parameters in each column is one, and chooses support points of 0 and 1. This model can be expressed equivalently as

    proc entropy pure data=m(obs=1) ;
        priors y1.x1 0 1 y1.x2 0 1 y1.x3 0 1 y1.x4 0 1;
        priors y2.x1 0 1 y2.x2 0 1 y2.x3 0 1 y2.x4 0 1;
        priors y3.x1 0 1 y3.x2 0 1 y3.x3 0 1 y3.x4 0 1;
        priors y4.x1 0 1 y4.x2 0 1 y4.x3 0 1 y4.x4 0 1;
        model y1 = x1-x4 / noint;
        model y2 = x1-x4 / noint;
        model y3 = x1-x4 / noint;
        model y4 = x1-x4 / noint;
        restrict y1.x1 + y2.x1 + y3.x1 + y4.x1 = 1;
        restrict y1.x2 + y2.x2 + y3.x2 + y4.x2 = 1;
        restrict y1.x3 + y2.x3 + y3.x3 + y4.x3 = 1;
        restrict y1.x4 + y2.x4 + y3.x4 + y4.x4 = 1;
    run;
The transition matrix is given in Figure 14.18.

**Figure 14.18** Estimate of \( P \) by Using One Transition

**Prior Distribution of Parameter \( T \)**

The **ENTROPY Procedure**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Estimate</th>
<th>Information Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1.x1</td>
<td>0.463407</td>
<td>0.0039</td>
</tr>
<tr>
<td>y1.x2</td>
<td>0.41055</td>
<td>0.0232</td>
</tr>
<tr>
<td>y1.x3</td>
<td>0.356272</td>
<td>0.0605</td>
</tr>
<tr>
<td>y1.x4</td>
<td>0.302163</td>
<td>0.1161</td>
</tr>
<tr>
<td>y2.x1</td>
<td>0.272755</td>
<td>0.1546</td>
</tr>
<tr>
<td>y2.x2</td>
<td>0.271459</td>
<td>0.1564</td>
</tr>
<tr>
<td>y2.x3</td>
<td>0.267252</td>
<td>0.1625</td>
</tr>
<tr>
<td>y2.x4</td>
<td>0.260084</td>
<td>0.1731</td>
</tr>
<tr>
<td>y3.x1</td>
<td>0.119926</td>
<td>0.4709</td>
</tr>
<tr>
<td>y3.x2</td>
<td>0.148481</td>
<td>0.3940</td>
</tr>
<tr>
<td>y3.x3</td>
<td>0.180224</td>
<td>0.3194</td>
</tr>
<tr>
<td>y3.x4</td>
<td>0.214394</td>
<td>0.2502</td>
</tr>
<tr>
<td>y4.x1</td>
<td>0.143903</td>
<td>0.4056</td>
</tr>
<tr>
<td>y4.x2</td>
<td>0.169504</td>
<td>0.3434</td>
</tr>
<tr>
<td>y4.x3</td>
<td>0.196252</td>
<td>0.2856</td>
</tr>
<tr>
<td>y4.x4</td>
<td>0.223364</td>
<td>0.2337</td>
</tr>
</tbody>
</table>

Note that \( P \) varies greatly from the true solution.

If two transitions are used instead (OBS=2), the resulting transition matrix is shown in Figure 14.19.

```plaintext
proc entropy markov pure data=m(obs=2);
    model y1-y4 = x1-x4;
run;
```
Figure 14.19  Estimate of $P$ by Using Two Transitions

Prior Distribution of Parameter $T$

This transition matrix is much closer to the actual transition matrix.

If, in addition to the transitions, you had other information about the transition matrix, such as your own company’s transition values, that information can be added as restrictions to the parameter estimates. For noisy data, the PURE option should be dropped. Note that this example has six zero probabilities in the transition matrix; the accurate estimation of transition matrices with fewer zero probabilities generally requires more transition observations.

Analyzing Multinomial Response Data

Multinomial discrete choice models suffer the same problems with collinearity of the regressors and small sample sizes as linear models. Unordered multinomial discrete choice models can be estimated using a variant of GME for discrete models called GME-D.

Consider the model shown in Golan, Judge, and Perloff (1996). In this model, there are five occupational categories, and the categories are considered a function of four individual characteristics. The sample contains 337 individuals.

```plaintext
data kpdata;
  input job x1 x2 x3 x4;
datalines;
  0 1 3 11 1
...
```

... more lines ...
The dependent variable in this data, job, takes on values 0 through 4. Support points are used only for the error terms; so error supports are specified on the MODEL statement.

```plaintext
proc entropy data=kpdata gmed tech=nra;
    model job = x1 x2 x3 x4 / noint
        esupports=( -.1 -0.0666 -0.0333 0 0.0333 0.0666 .1 );
run;
```

**Figure 14.20** Estimate of Jobs Model by Using GME-D

Prior Distribution of Parameter T

### The ENTROPY Procedure

| Variable | Estimate | Approx Std Err | Approx t Value | Approx Pr > |t|
|----------|----------|----------------|----------------|-------------|
| x1_1     | 1.802572 | 1.3610         | 1.32           | 0.1863      |
| x2_1     | -0.00251 | 0.0154         | -0.16          | 0.8705      |
| x3_1     | -0.17282 | 0.0885         | -1.95          | 0.0517      |
| x4_1     | 1.054659 | 0.6986         | 1.51           | 0.1321      |
| x1_2     | 0.089156 | 1.2764         | 0.07           | 0.9444      |
| x2_2     | 0.019947 | 0.0146         | 1.37           | 0.1718      |
| x3_2     | 0.010716 | 0.0830         | 0.13           | 0.8974      |
| x4_2     | 0.288629 | 0.5775         | 0.50           | 0.6176      |
| x1_3     | -4.62047 | 1.6476         | -2.80          | 0.0053      |
| x2_3     | 0.026175 | 0.0166         | 1.58           | 0.1157      |
| x3_3     | 0.245198 | 0.0986         | 2.49           | 0.0134      |
| x4_3     | 1.285466 | 0.8367         | 1.54           | 0.1254      |
| x1_4     | -9.72734 | 1.5813         | -6.15          | <0.0001     |
| x2_4     | 0.027382 | 0.0156         | 1.75           | 0.0805      |
| x3_4     | 0.660836 | 0.0947         | 6.98           | <0.0001     |
| x4_4     | 1.47479  | 0.6970         | 2.12           | 0.0351      |

Note there are five estimates of the parameters produced for each regressor, one for each choice. The first choice is restricted to zero for normalization purposes. PROC ENTROPY drops the zeroed regressors. PROC ENTROPY also generates tables of marginal effects for each regressor. The following statements generate the marginal effects table for the previous analysis at the means of the variables.

```plaintext
proc entropy data=kpdata gmed tech=nra;
    model job = x1 x2 x3 x4 / noint
        esupports=( -.1 -0.0666 -0.0333 0 0.0333 0.0666 .1 )
        marginals;
run;
```
**Figure 14.21** Estimate of Jobs Model by Using GME-D (Marginals)

**Prior Distribution of Parameter $T$**

The ENTROPY Procedure

<table>
<thead>
<tr>
<th>Variable</th>
<th>Marginal Effect</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1_0</td>
<td>0.338758</td>
<td>1</td>
</tr>
<tr>
<td>x2_0</td>
<td>-0.0019</td>
<td>20.50148</td>
</tr>
<tr>
<td>x3_0</td>
<td>-0.02129</td>
<td>13.09496</td>
</tr>
<tr>
<td>x4_0</td>
<td>-0.09917</td>
<td>0.916914</td>
</tr>
<tr>
<td>x1_1</td>
<td>0.859883</td>
<td>1</td>
</tr>
<tr>
<td>x2_1</td>
<td>-0.00345</td>
<td>20.50148</td>
</tr>
<tr>
<td>x3_1</td>
<td>-0.0648</td>
<td>13.09496</td>
</tr>
<tr>
<td>x4_1</td>
<td>0.034396</td>
<td>0.916914</td>
</tr>
<tr>
<td>x1_2</td>
<td>0.86101</td>
<td>1</td>
</tr>
<tr>
<td>x2_2</td>
<td>0.000963</td>
<td>20.50148</td>
</tr>
<tr>
<td>x3_2</td>
<td>-0.04948</td>
<td>13.09496</td>
</tr>
<tr>
<td>x4_2</td>
<td>-0.16297</td>
<td>0.916914</td>
</tr>
<tr>
<td>x1_3</td>
<td>-0.25969</td>
<td>1</td>
</tr>
<tr>
<td>x2_3</td>
<td>0.0015</td>
<td>20.50148</td>
</tr>
<tr>
<td>x3_3</td>
<td>0.009289</td>
<td>13.09496</td>
</tr>
<tr>
<td>x4_3</td>
<td>0.065569</td>
<td>0.916914</td>
</tr>
<tr>
<td>x1_4</td>
<td>-1.79996</td>
<td>1</td>
</tr>
<tr>
<td>x2_4</td>
<td>0.00288</td>
<td>20.50148</td>
</tr>
<tr>
<td>x3_4</td>
<td>0.126283</td>
<td>13.09496</td>
</tr>
<tr>
<td>x4_4</td>
<td>0.162172</td>
<td>0.916914</td>
</tr>
</tbody>
</table>

The marginals are derivatives of the probabilities with respect to each variable and so summarize how a small change in each variable affects the overall probability.

PROC ENTROPY also enables the user to specify where the derivative is evaluated, as shown below:

```plaintext
proc entropy data=kpdata gmed tech=nra;
    model job = x1 x2 x3 x4 / noint
        esupports=( -.1 -0.0666 -0.0333 0 0.0333 0.0666 .1 )
        marginals=( x2=.4 x3=10 x4=0);
run;
```
In this example, you evaluate the derivative when $x_1=1$, $x_2=0.4$, $x_3=10$, and $x_4=0$. If the user neglects a variable, PROC ENTROPY uses its mean value.
Syntax: ENTROPY Procedure

The following statements can be used with the ENTROPY procedure:

```plaintext
PROC ENTROPY options ;
   BOUNDS bound1 < , bound2, . . . > ;
   BY variable < variable . . . > ;
   ID variable < variable . . . > ;
   MODEL variable = variable < variable . . . > ;
   PRIORS variable < support points > variable < value > . . . ;
   RESTRICT restriction1 < , restriction2 . . . > ;
   TEST < "name" > test1 < , test2 . . . > </ options > ;
   WEIGHT variable ;
```

Functional Summary

The statements and options in the ENTROPY procedure are summarized in the following table.

<table>
<thead>
<tr>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Set Options</strong></td>
<td>ENTROPY</td>
<td></td>
</tr>
<tr>
<td>specify the input data set for the variables</td>
<td>DATA=</td>
<td></td>
</tr>
<tr>
<td>specify the input data set for support points and priors</td>
<td>PDATA=</td>
<td></td>
</tr>
<tr>
<td>specify the output data set for residual, predicted, and actual values</td>
<td>OUT=</td>
<td></td>
</tr>
<tr>
<td>specify the output data set for the support points and priors</td>
<td>OUTF=</td>
<td></td>
</tr>
<tr>
<td>write the covariance matrix of the estimates to OUTTEST= data set</td>
<td>OUTCOV</td>
<td></td>
</tr>
<tr>
<td>write the parameter estimates to a data set</td>
<td>OUTTEST=</td>
<td></td>
</tr>
<tr>
<td>write the Lagrange multiplier estimates to a data set</td>
<td>OUTL=</td>
<td></td>
</tr>
<tr>
<td>write the covariance matrix of the equation errors to a data set</td>
<td>OUTS=</td>
<td></td>
</tr>
<tr>
<td>write the S matrix used in the objective function definition to a data set</td>
<td>OUTSUSED=</td>
<td></td>
</tr>
<tr>
<td>read the covariance matrix of the equation errors</td>
<td>SDATA=</td>
<td></td>
</tr>
<tr>
<td><strong>Printing Options</strong></td>
<td>ENTROPY</td>
<td></td>
</tr>
<tr>
<td>request that the procedure produce graphics via the Output Delivery System</td>
<td>PLOTS=</td>
<td></td>
</tr>
<tr>
<td>print collinearity diagnostics</td>
<td>COLLIN</td>
<td></td>
</tr>
<tr>
<td>suppress the normal printed output</td>
<td>NOPRINT</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Statement</td>
<td>Option</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td><strong>Options to Control Iteration Output</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>print a summary iteration listing</td>
<td>ENTROPY</td>
<td>ITPRINT</td>
</tr>
<tr>
<td><strong>Options to Control the Minimization Process</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>specify the convergence criteria</td>
<td>ENTROPY</td>
<td>CONVERGE=</td>
</tr>
<tr>
<td>specify the maximum number of iterations allowed</td>
<td>ENTROPY</td>
<td>MAXITER=</td>
</tr>
<tr>
<td>specify the maximum number of subiterations allowed</td>
<td>ENTROPY</td>
<td>MAXSUBITER=</td>
</tr>
<tr>
<td>select the iterative minimization method to use</td>
<td>ENTROPY</td>
<td>METHOD=</td>
</tr>
<tr>
<td><strong>Statements That Declare Variables</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>specify BY-group processing</td>
<td>BY</td>
<td></td>
</tr>
<tr>
<td>specify a weight variable</td>
<td>WEIGHT</td>
<td></td>
</tr>
<tr>
<td>specify identifying variables</td>
<td>ID</td>
<td></td>
</tr>
<tr>
<td><strong>General PROC ENTROPY Statement Options</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>specify seemingly unrelated regression</td>
<td>ENTROPY</td>
<td>SUR</td>
</tr>
<tr>
<td>specify iterated seemingly unrelated regression</td>
<td>ENTROPY</td>
<td>ITSUR</td>
</tr>
<tr>
<td>specify data-constrained generalized maximum entropy</td>
<td>ENTROPY</td>
<td>GME</td>
</tr>
<tr>
<td>specify moment generalized maximum entropy</td>
<td>ENTROPY</td>
<td>GMEM</td>
</tr>
<tr>
<td>specify the denominator for computing variances and covariances</td>
<td>ENTROPY</td>
<td>VARDEF=</td>
</tr>
<tr>
<td><strong>General TEST Statement Options</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>specify that a Wald test be computed</td>
<td>TEST</td>
<td>WALD</td>
</tr>
<tr>
<td>specify that a Lagrange multiplier test be computed</td>
<td>TEST</td>
<td>LM</td>
</tr>
<tr>
<td>specify that a likelihood ratio test be computed</td>
<td>TEST</td>
<td>LR</td>
</tr>
<tr>
<td>request all three types of tests</td>
<td>TEST</td>
<td>ALL</td>
</tr>
</tbody>
</table>
PROC ENTROPY Statement

PROC ENTROPY options ;

The following options can be specified in the PROC ENTROPY statement.

General Options

COLLIN
requests that the collinearity diagnostics of the $X'X$ matrix be printed.

COVBEST=CROSS | GME | GMEM
specifies the method for producing the covariance matrix of parameters for output and for standard error calculations. GMEM and GME are aliases and are the default.

GME | GCE
requests generalized maximum entropy or generalized cross entropy. This is the default estimation method.

GMEM | GCEM
requests moment maximum entropy or the moment cross entropy.

GMED
requests a variant of GME suitable for multinomial discrete choice models.

MARKOV
specifies that the model is a first-order Markov model.

PURE
specifies a regression without an error term.

SUR | ITSUR
specifies seemingly unrelated regression or iterated seemingly unrelated regression.

VARDEF=N | WGT | DF | WDF
specifies the denominator to be used in computing variances and covariances. VARDEF=N specifies that the number of nonmissing observations be used. VARDEF=WGT specifies that the sum of the weights be used. VARDEF=DF specifies that the number of nonmissing observations minus the model degrees of freedom (number of parameters) be used. VARDEF=WDF specifies that the sum of the weights minus the model degrees of freedom be used. The default is VARDEF=DF.

Data Set Options

DATA=SAS-data-set
specifies the input data set. Values for the variables in the model are read from this data set.

PDATA=SAS-data-set
names the SAS data set that contains the data about priors and supports.
OUT=SAS-data-set
names the SAS data set to contain the residuals from each estimation.

OUTCOV
COVOUT
writes the covariance matrix of the estimates to the OUTEST= data set in addition to the parameter estimates. The OUTCOV option is applicable only if the OUTEST= option is also specified.

OUTEST=SAS-data-set
names the SAS data set to contain the parameter estimates and optionally the covariance of the estimates.

OUTL=SAS-data-set
names the SAS data set to contain the estimated Lagrange multipliers for the models.

OUTP=SAS-data-set
names the SAS data set to contain the support points and estimated probabilities.

OUTS=SAS-data-set
names the SAS data set to contain the estimated covariance matrix of the equation errors. This is the covariance of the residuals computed from the parameter estimates.

OUTSUSED=SAS-data-set
names the SAS data set to contain the S matrix used in the objective function definition. The OUTSUSED= data set is the same as the OUTS= data set for the methods that iterate the S matrix.

SDATA=SAS-data-set
specifies a data set that provides the covariance matrix of the equation errors. The matrix read from the SDATA= data set is used for the equation error covariance matrix (S matrix) in the estimation. The SDATA= matrix is used to provide only the initial estimate of S for the methods that iterate the S matrix.

Printing Options

ITPRINT
prints the parameter estimates, objective function value, and convergence criteria at each iteration.

NOPRINT
suppresses the normal printed output but does not suppress error listings. Using any other print option turns the NOPRINT option off.

PLOTS=global-plot-options | plot-request
controls the plots that the ENTROPY procedure produces. (For general information about ODS Graphics, see Chapter 21, “Statistical Graphics Using ODS” (SAS/STAT User’s Guide).) The global-plot-options apply to all relevant plots generated by the ENTROPY procedure.

The global-plot-options supported by the ENTROPY procedure are as follows:

ONLY suppresses the default plots. Only the plots specifically requested are produced.
UNPACKPANEL displays each graph separately. (By default, some graphs can appear together in a single panel.)
The specific `plot-request` values supported by the ENTROPY procedure are as follows:

- **ALL** requests that all plots appropriate for the particular analysis be produced. ALL is equivalent to specifying FITPLOT, COOKSD, QQ, RESIDUALHISTOGRAM, and STUDENTRESIDUAL.
- **FITPLOT** plots the predicted and actual values.
- **COOKSD** produces the Cook’s D plot.
- **QQ** produces a Q-Q plot of residuals.
- **RESIDUALHISTOGRAM** plots the histogram of residuals.
- **STUDENTRESIDUAL** plots the studentized residuals.
- **NONE** suppresses all plots.

The default behavior is to plot all plots appropriate for the particular analysis (ALL) in a panel.

### Options to Control the Minimization Process

The following options can be helpful if a convergence problem occurs for a given model and set of data. The ENTROPY procedure uses the nonlinear optimization subsystem (NLO) to perform the model optimizations. In addition to the options listed below, all options supported in the NLO subsystem can be specified on the ENTROPY procedure statement. See Chapter 7, “Nonlinear Optimization Methods,” for more details.

- **CONVERGE**=`value`
  - Specifies the convergence criteria for S-iterated methods. The convergence measure computed during model estimation must be less than `value` before convergence is assumed. The default value is `CONVERGE=0.001`.

- **DUAL | PRIMAL**
  - Specifies whether the optimization problem is solved using the dual or primal form. The dual form is the default.

- **MAXITER**=`n`
  - Specifies the maximum number of iterations allowed. The default is `MAXITER=100`.

- **MAXSUBITER**=`n`
  - Specifies the maximum number of subiterations allowed for an iteration. The `MAXSUBITER` option limits the number of step halvings. The default is `MAXSUBITER=30`.

- **METHOD**=`TR | NEWRAP | NRR | QN | CONGR | NSIMP | DBLDOG | LEVMAR`

- **TECHNIQUE**=`TR | NEWRAP | NRR | QN | CONGR | NSIMP | DBLDOG | LEVMAR`

- **TECH**=`TR | NEWRAP | NRR | QN | CONGR | NSIMP | DBLDOG | LEVMAR`
  - Specifies the iterative minimization method to use. METHOD=TR specifies the trust region method, METHOD=NEWRAP specifies the Newton-Raphson method, METHOD=NRR specifies the Newton-Raphson ridge method, and METHOD=QN specifies the quasi-Newton method. See Chapter 7, “Nonlinear Optimization Methods,” for more details about optimization methods. The default is METHOD=QN for the dual form and METHOD=NEWRAP for the primal form.
The BOUNDS statement imposes simple boundary constraints on the parameter estimates. BOUNDS statement constraints refer to the parameters estimated by the ENTROPY procedure. You can specify any number of BOUNDS statements.

Each boundary constraint is composed of variables, constants, and inequality operators in the following form:

\[ \text{item operator item } <, \text{operator item } <, \text{operator item } \ldots \geq \]

Each item is a constant, the name of a regressor variable, or a list of regressor names. Each operator is <, >, <=, or >=.

You can use either the BOUNDS statement or the RESTRICT statement to impose boundary constraints; the BOUNDS statement provides a simpler syntax for specifying inequality constraints. See section “RESTRICT Statement” on page 817 for more information about the computational details of estimation with inequality restrictions.

Lagrange multipliers are reported for all the active boundary constraints. In the printed output and in the OUTEST= data set, the Lagrange multiplier estimates are identified with the names BOUND1, BOUND2, and so forth. The probability of the Lagrange multipliers are computed using a beta distribution (LaMotte 1994). Nonactive or nonbinding bounds have no effect on the estimation results and are not noted in the output. To give the constraints more descriptive names, use the RESTRICT statement instead of the BOUNDS statement.

The following BOUNDS statement constrains the estimates of the coefficients of WAGE and TARGET and the 10 coefficients of x1 through x10 to be between zero and one. This example illustrates the use of parameter lists to specify boundary constraints.

\[
\text{bounds } 0 < \text{wage target x1-x10 } \leq 1; \\
\]

The following is an example of the use of the BOUNDS statement to impose boundary constraints on the variables X1, X2, and X3:

```
proc entropy data=zero;
    bounds .1 <= x1 <= 100,
    0 <= x2 <= 25.6,
    0 <= x3 <= 5;
    model y = x1 x2 x3;
run;
```

The parameter estimates from this run are shown in Figure 14.23.
Figure 14.23 Output from Bounded Estimation

Prior Distribution of Parameter T

The ENTROPY Procedure

Variables(Supports(Weights)) x1 x2 x3 Intercept
Equations(Supports(Weights)) y

Prior Distribution of Parameter T

The ENTROPY Procedure
GME Estimation Summary

<table>
<thead>
<tr>
<th>Data Set Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA= WORK.ZERO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Minimization Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters Estimated 4</td>
</tr>
<tr>
<td>Covariance Estimator GME</td>
</tr>
<tr>
<td>Entropy Type Shannon</td>
</tr>
<tr>
<td>Entropy Form Dual</td>
</tr>
<tr>
<td>Numerical Optimizer Newton-Raphson</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Final Information Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective Function Value 6.292861</td>
</tr>
<tr>
<td>Signal Entropy 6.375715</td>
</tr>
<tr>
<td>Noise Entropy -0.08285</td>
</tr>
<tr>
<td>Normed Entropy (Signal) 0.990364</td>
</tr>
<tr>
<td>Normed Entropy (Noise) 1.004172</td>
</tr>
<tr>
<td>Parameter Information Index 0.009636</td>
</tr>
<tr>
<td>Error Information Index -0.00417</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Observations Processed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read 20</td>
</tr>
<tr>
<td>Used 20</td>
</tr>
</tbody>
</table>

NOTE: At GME iteration 20 convergence criteria met.

GME Summary of Residual Errors

<table>
<thead>
<tr>
<th>Equation</th>
<th>DF Model Error</th>
<th>SSE</th>
<th>MSE</th>
<th>Root MSE</th>
<th>R-Square</th>
<th>Adj R Sq</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>4</td>
<td>16</td>
<td>83281.0</td>
<td>288.6</td>
<td>-0.0013</td>
<td>-0.1891</td>
</tr>
</tbody>
</table>
BY Statement

BY variables;

A BY statement is used to obtain separate estimates for observations in groups defined by the BY variables. To save parameter estimates for each BY group, use the OUTEST= option.

ID Statement

ID variables;

The ID statement specifies variables to identify observations in error messages or other listings and in the OUT= data set. The ID variables are normally SAS date or datetime variables. If more than one ID variable is used, the first variable is used to identify the observations and the remaining variables are added to the OUT= data set.

MODEL Statement

MODEL dependent = regressors < / options >;

The MODEL statement specifies the dependent variable and independent regressor variables for the regression model. If no independent variables are specified in the MODEL statement, only the mean (intercept) is estimated. To model a system of equations, specify more than one MODEL statement.

The following options can be used in the MODEL statement after a slash (/).

ESUPPORTS=( support (prior) . . . )

specifies the support points and prior weights on the residuals for the specified equation. The default is the following five support values:

\[-10 \ast value, \ast value, 0, \ast value, 10 \ast value\]

where value is computed as

\[value = (max(y) - \bar{y}) \ast multiplier\]

---

**Figure 14.23  continued**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Estimate</th>
<th>Approx Std Err</th>
<th>t Value</th>
<th>Approx Pr &gt;</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0.1</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>3.33E-16</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>-0.00432</td>
<td>3.406E-6</td>
<td>-1269.3</td>
<td>&lt;.0001</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.25731</td>
<td>9130.3</td>
<td>0.00</td>
<td>0.9999</td>
<td>0 &lt;= x1</td>
</tr>
<tr>
<td></td>
<td>0.009384</td>
<td>0</td>
<td>-</td>
<td>0 &lt;= x2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.000025</td>
<td>0</td>
<td>-</td>
<td>0 &lt;= x3</td>
<td></td>
</tr>
</tbody>
</table>
for GME, where $y$ is the dependent variable, and

$$value = (\max(y) - \hat{y}) * multiplier * nobs * \max(X) * 0.1$$

for generalized maximum entropy—moments (GME-M), where $X$ is the information matrix, and $nobs$ is the number of observations. The $multiplier$ depends on the MULTIPLIER= option. The MULTIPLIER= option defaults to 2 for unrestricted models and to 4 for restricted models. The prior probabilities default to the following:

$$0.0005, 0.333, 0.333, 0.333, 0.0005$$

The support points and prior weights are selected so that hypothesis tests can be performed without adding significant bias to the estimation. These prior probability values are ad hoc.

**NOINT**

suppresses the intercept parameter.

**MARGINALS = ( variable = value, . . . , variable = value)**

requests that the marginal effects of each variable be calculated for GME-D. Specifying the MARGINALS option with an optional list of values calculates the marginals at that vector of values. For example, if $x_1$–$x_4$ are explanatory variables, then including

$$\text{MARGINALS} = (x_1 = 2, x_2 = 4, x_3 = -1, x_4 = 5)$$

calculates the marginal effects at that vector. A skipped variable implies that its mean value is to be used.

**CENSORED ( ( UB | LB) = (variable | value ), ESUPPORTS = ( support (prior) . . . ) )**

specifies that the dependent variable be observed with censoring and specifies the censoring thresholds and the supports of the censored observations.

**CATEGORY= variable**

specifies the variable that keeps track of the categories the dependent variable is in when there is range censoring. When the actual value is observed, this variable should be set to MISSING.

**RANGE ( ID = (QS | INT) L = ( NUMBER ) R = ( NUMBER ) , ESUPPORTS = ( support < (prior) . . . ) )**

specifies that the dependent variable be range bound. The RANGE option defines the range and the key (RANGE) that is used to identify the observation as being range bound. The RANGE = value should be some value in the CATEGORY= variable. The L and R define, respectively, the left endpoint of the range and the right endpoint of the range. ESUPPORTS sets the error supports on the variable.

---

### PRIORS Statement

**PRIORS variable < support points < (priors) >> variable < support points < (priors) >> . . . ;**

The PRIORS statement specifies the support points and prior weights for the coefficients on the variables.

Support points for coefficients default to five points, determined as follows:

$$-2 * value, -value, 0, value, 2 * value$$
where value is computed as

\[ \text{value} = (||\text{mean}|| + 3 \times \text{stderr}) \times \text{multiplier} \]

where the mean and the stderr are obtained from OLS and the multiplier depends on the MULTIPLIER= option. The MULTIPLIER= option defaults to 2 for unrestricted models and to 4 for restricted models. The prior probabilities for each support point default to the uniform distribution.

The number of support points must be at least two. If priors are specified, they must be positive and there must be the same number of priors as there are support points. Priors and support points can also be specified through the PDATA= data set.

---

**RESTRICT Statement**

RESTRICT restriction1 < , restriction2 ... > ;

The RESTRICT statement is used to impose linear restrictions on the parameter estimates. You can specify any number of RESTRICT statements.

Each restriction is written as an optional name, followed by an expression, followed by an equality operator (=) or an inequality operator (<, >, <=, >=), followed by a second expression:

```
<"name" > expression operator expression
```

where "name" is a string used to identify the restriction in the printed output and in the OUTEST= data set. The operator can be =, <, >, <=, or >=. The operator and second expression are optional, as in the TEST statement, where they default to = 0.

Restriction expressions can be composed of variable names, multiplication (*), and addition (+) operators, and constants. Variable names in restriction expressions must be among the variables whose coefficients are estimated by the model. The restriction expressions must be a linear function of the variables.

The following is an example of the use of the RESTRICT statement:

```plaintext
proc entropy data=one;
   restrict y1.x1*2 <= x2 + y2.x1;
   model y1 = x1 x2;
   model y2 = x1 x3;
run;
```

This example illustrates the use of compound names, y1.x1, to specify coefficients of specific equations.

---

**TEST Statement**

TEST < "name" > test1 < , test2 ... > < , / options > ;

The TEST statement performs tests of linear hypotheses on the model parameters.

The TEST statement applies only to parameters estimated in the model. You can specify any number of TEST statements.
Each test is written as an expression optionally followed by an equal sign (=) and a second expression:

```
expression  <=  expression
```

Test expressions can be composed of variable names, multiplication (*), addition (+), and subtraction (−) operators, and constants. Variables named in test expressions must be among the variables estimated by the model.

If you specify only one expression in a TEST statement, that expression is tested against zero. For example, the following two TEST statements are equivalent:

```
test a + b;

test a + b = 0;
```

When you specify multiple tests on the same TEST statement, a joint test is performed. For example, the following TEST statement tests the joint hypothesis that both of the coefficients on a and b are equal to zero:

```
test a, b;
```

To perform separate tests rather than a joint test, use separate TEST statements. For example, the following TEST statements test the two separate hypotheses that a is equal to zero and that b is equal to zero:

```
test a;

test b;
```

You can use the following options in the TEST statement:

- **WALD** specifies that a Wald test be computed. WALD is the default.
- **LM**
- **RAO**
- **LAGRANGE** specifies that a Lagrange multiplier test be computed.
- **LR**
- **LIKE** specifies that a pseudo-likelihood ratio test be computed.
- **ALL** requests all three types of tests.
- **OUT=** specifies the name of an output SAS data set that contains the test results. The format of the OUT= data set produced by the TEST statement is similar to that of the OUTEST= data set.
WEIGHT Statement

WEIGHT variable;

The WEIGHT statement specifies a variable to supply weighting values to use for each observation in estimating parameters.

If the weight of an observation is nonpositive, that observation is not used for the estimation. Variable must be a numeric variable in the input data set. The regressors and the dependent variables are multiplied by the square root of the weight variable to form the weighted X matrix and the weighted dependent variable. The same weight is used for all MODEL statements.

Details: ENTROPY Procedure

Shannon’s measure of entropy for a distribution is given by

\[
\max \sum_{i=1}^{n} p_i \ln(p_i)
\]

subject to \( \sum_{i=1}^{n} p_i = 1 \)

where \( p_i \) is the probability associated with the \( i \)th support point. Properties that characterize the entropy measure are set forth by Kapur and Kesavan (1992).

The objective is to maximize the entropy of the distribution with respect to the probabilities \( p_i \) and subject to constraints that reflect any other known information about the distribution (Jaynes 1957). This measure, in the absence of additional information, reaches a maximum when the probabilities are uniform. A distribution other than the uniform distribution arises from information already known.

Generalized Maximum Entropy

Reparameterization of the errors in a regression equation is the process of specifying a support for the errors, observation by observation. If a two-point support is used, the error for the \( t \)th observation is reparameterized by setting \( e_t = w_{t1} v_{t1} + w_{t2} v_{t2} \), where \( v_{t1} \) and \( v_{t2} \) are the upper and lower bounds for the \( t \)th error \( e_t \), and \( w_{t1} \) and \( w_{t2} \) represent the weight associated with the point \( v_{t1} \) and \( v_{t2} \). The error distribution is usually chosen to be symmetric, centered around zero, and the same across observations so that \( v_{t1} = -v_{t2} = R \), where \( R \) is the support value chosen for the problem (Golan, Judge, and Miller 1996).

The generalized maximum entropy (GME) formulation was proposed for the ill-posed or underdetermined case where there is insufficient data to estimate the model with traditional methods. \( \beta \) is reparameterized by defining a support for \( \beta \) (and a set of weights in the cross entropy case), which defines a prior distribution for \( \beta \).

In the simplest case, each \( \beta_k \) is reparameterized as \( \beta_k = p_{k1} z_{k1} + p_{k2} z_{k2} \), where \( p_{k1} \) and \( p_{k2} \) represent the probabilities ranging from \([0,1]\) for each \( \beta \), and \( z_{k1} \) and \( z_{k2} \) represent the lower and upper bounds placed...
on $\beta_k$. The support points, $z_{k1}$ and $z_{k2}$, are usually distributed symmetrically around the most likely value for $\beta_k$ based on some prior knowledge.

With these reparameterizations, the GME estimation problem is

$$\text{maximize } H(p, w) = -p' \ln(p) - w' \ln(w)$$

subject to

$$y = XZp + Vw$$

$$1_K = (I_K \otimes 1'_L) p$$

$$1_T = (I_T \otimes 1'_L) w$$

where $y$ denotes the column vector of length $T$ of the dependent variable; $X$ denotes the $(T \times K)$ matrix of observations of the independent variables; $p$ denotes the $LK$ column vector of weights associated with the points in $Z$; $w$ denotes the $LT$ column vector of weights associated with the points in $V$; $1_K$, $1_L$, and $1_T$ are $K$-, $L$-, and $T$-dimensional column vectors, respectively, of ones; and $I_K$ and $I_T$ are $(K \times K)$ and $(T \times T)$ dimensional identity matrices.

These equations can be rewritten using set notation as follows:

$$\text{maximize } H(p, w) = -\sum_{l=1}^{L} \sum_{k=1}^{K} p_{kl} \ln(p_{kl}) - \sum_{l=1}^{L} \sum_{t=1}^{T} w_{tl} \ln(w_{tl})$$

subject to

$$y_t = \sum_{l=1}^{L} \left[ \sum_{k=1}^{K} (X_{kt} Z_{kl} p_{kl}) + V_{tl} w_{tl} \right]$$

$$\sum_{l=1}^{L} p_{kl} = 1 \text{ and } \sum_{l=1}^{L} w_{tl} = 1$$

The subscript $l$ denotes the support point ($l=1, 2, ..., L$), $k$ denotes the parameter ($k=1, 2, ..., K$), and $t$ denotes the observation ($t=1, 2, ..., T$).

The GME objective is strictly concave; therefore, a unique solution exists. The optimal estimated probabilities, $p$ and $w$, and the prior supports, $Z$ and $V$, can be used to form the point estimates of the unknown parameters, $\beta$, and the unknown errors, $e$.

---

**Generalized Cross Entropy**

Kullback and Leibler (1951) cross entropy measures the “discrepancy” between one distribution and another. Cross entropy is called a measure of discrepancy rather than distance because it does not satisfy some of the properties one would expect of a distance measure. (See Kapur and Kesavan (1992) for a discussion of cross entropy as a measure of discrepancy.) Mathematically, cross entropy is written as

$$\text{minimize } \sum_{i=1}^{n} p_i \ln( p_i / q_i )$$

subject to

$$\sum_{i=1}^{n} p_i = 1,$$
Generalized Cross Entropy

where \( q_i \) is the probability associated with the \( i \)th point in the distribution from which the discrepancy is measured. The \( q_i \) (in conjunction with the support) are often referred to as the prior distribution. The measure is nonnegative and is equal to zero when \( p_i \) equals \( q_i \). The properties of the cross entropy measure are examined by Kapur and Kesavan (1992).

The principle of minimum cross entropy (Kullback 1959; Good 1963) states that one should choose probabilities that are as close as possible to the prior probabilities. That is, out of all probability distributions that satisfy a given set of constraints which reflect known information about the distribution, choose the distribution that is closest (as measured by \( p(\ln(p) - \ln(q)) \)) to the prior distribution. When the prior distribution is uniform, maximum entropy and minimum cross entropy produce the same results (Kapur and Kesavan 1992), where the higher values for entropy correspond exactly with the lower values for cross entropy.

If the prior distributions are nonuniform, the problem can be stated as a generalized cross entropy (GCE) formulation. The cross entropy terminology specifies weights, \( q_i \) and \( u_i \), for the points \( Z \) and \( V \), respectively. Given informative prior distributions on \( Z \) and \( V \), the GCE problem is

\[
\begin{align*}
\text{minimize} \quad I(p,q,w,u) &= p' \ln(p/q) + w' \ln(w/u) \\
\text{subject to} \quad y &= XZp + Vw \\
1_K &= (I_K \otimes 1_L) p \\
1_T &= (I_T \otimes 1_T') w
\end{align*}
\]

where \( y \) denotes the \( T \) column vector of observations of the dependent variables; \( X \) denotes the \((T \times K)\) matrix of observations of the independent variables; \( q \) and \( p \) denote \( L \times K \) column vectors of prior and posterior weights, respectively, associated with the points in \( Z \); \( u \) and \( w \) denote the \( L \times T \) column vectors of prior and posterior weights, respectively, associated with the points in \( V \); \( 1_K, 1_L, \) and \( 1_T \) are \( K-, L-, \) and \( T\)-dimensional column vectors, respectively, of ones; and \( I_K \) and \( I_T \) are \((K \times K)\) and \((T \times T)\) dimensional identity matrices.

The optimization problem can be rewritten using set notation as follows

\[
\begin{align*}
\text{minimize} \quad I(p,q,w,u) &= \sum_{l=1}^{L} \sum_{k=1}^{K} p_{kl} \ln(p_{kl}/q_{kl}) + \sum_{l=1}^{L} \sum_{t=1}^{T} w_{tl} \ln(w_{tl}/u_{tl}) \\
\text{subject to} \quad y_t &= \sum_{l=1}^{L} \left[ \sum_{k=1}^{K} (X_{kt} Z_{kl} p_{kl}) + V_{tl} w_{tl} \right] \\
\sum_{l=1}^{L} p_{kl} &= 1 \quad \text{and} \quad \sum_{l=1}^{L} w_{tl} = 1
\end{align*}
\]

The subscript \( l \) denotes the support point \((l=1, 2, ..., L)\), \( k \) denotes the parameter \((k=1, 2, ..., K)\), and \( t \) denotes the observation \((t=1, 2, ..., T)\).

The objective function is strictly convex; therefore, there is a unique global minimum for the problem (Golan, Judge, and Miller 1996). The optimal estimated weights, \( p \) and \( w \), and the prior supports, \( Z \) and \( V \), can be used to form the point estimates of the unknown parameters, \( \beta \), and the unknown errors, \( e \), by using
\[ \beta = Zp = \begin{bmatrix} z_{11} & \cdots & z_{L1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & z_{12} & \cdots & z_{L2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & z_{1K} & \cdots & z_{LK} \end{bmatrix} \]

\[ e = Vw = \begin{bmatrix} v_{11} & \cdots & v_{L1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & v_{12} & \cdots & v_{L2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{1T} & \cdots & v_{LT} \end{bmatrix} \]

**Computational Details**

This constrained estimation problem can be solved either directly (primal) or by using the dual form. Either way, it is prudent to factor out one probability for each parameter and each observation as the sum of the other probabilities. This factoring reduces the computational complexity significantly. If the primal formalization is used and two support points are used for the parameters and the errors, the resulting GME problem is \( O(n\text{parms} \cdot n\text{obs})^3 \). For the dual form, the problem is \( O(n\text{obs})^3 \). Therefore for large data sets, GME-M should be used instead of GME.

**Moment Generalized Maximum Entropy**

The default estimation technique is moment generalized maximum entropy (GME-M). This is simply GME with the data constraints modified by multiplying both sides by \( X' \). GME-M then becomes

\[
\underset{\text{maximize}}{H(p, w) = -p' \ln(p) - w' \ln(w)} \\
\text{subject to} \\
X'y = X'XZp + X'Vw \\
1_K = (I_K \otimes 1_L')p \\
1_T = (I_T \otimes 1_L')w
\]
There is also the cross entropy version of GME-M, which has the same form as GCE but with the moment constraints.

**GME versus GME-M**

GME-M is more computationally attractive than GME for large data sets because the computational complexity of the estimation problem depends primarily on the number of parameters and not on the number of observations. GME-M is based on the first moment of the data, whereas GME is based on the data itself. If the distribution of the residuals is well defined by its first moment, then GME-M is a good choice. So if the residuals are normally distributed or exponentially distributed, then GME-M should be used. On the other hand if the distribution is Cauchy, lognormal, or some other distribution where the first moment does not describe the distribution, then use GME. See Example 14.1 for an illustration of this point.

---

**Maximum Entropy-Based Seemingly Unrelated Regression**

In a multivariate regression model, the errors in different equations might be correlated. In this case, the efficiency of the estimation can be improved by taking these cross-equation correlations into account. Seemingly unrelated regression (SUR), also called joint generalized least squares (JGLS) or Zellner estimation, is a generalization of OLS for multi-equation systems.

Like SUR in the least squares setting, the generalized maximum entropy SUR (GME-SUR) method assumes that all the regressors are independent variables and uses the correlations among the errors in different equations to improve the regression estimates. The GME-SUR method requires an initial entropy regression to compute residuals. The entropy residuals are used to estimate the cross-equation covariance matrix. In the iterative GME-SUR (ITGME-SUR) case, the preceding process is repeated by using the residuals from the GME-SUR estimation to estimate a new cross-equation covariance matrix. ITGME-SUR method alternates between estimating the system coefficients and estimating the cross-equation covariance matrix until the estimated coefficients and covariance matrix converge.

The estimation problem becomes the generalized maximum entropy system adapted for multi-equations as follows:

\[
\text{maximize} \quad H(p, w) = -p' \ln(p) - w' \ln(w) \\
\text{subject to} \quad y = XZp + Vw \\
1_{KM} = (I_{KM} \otimes 1'_L)p \\
1_{MT} = (I_{MT} \otimes 1'_L)w
\]

where

\[
\beta = Zp
\]
Using this notation, the maximum entropy problem that is analogous to the OLS problem used as the initial step of the traditional SUR approach is

$$\begin{align*}
\text{maximize} \quad & H(p, w) = -p' \ln(p) - w' \ln(w) \\
\text{subject to} \quad & \begin{array}{l}
y - X Z p = \sqrt{\Sigma} V w \\
1_{KM} = (I_{KM} \otimes 1_{L}) p \\
1_{MT} = (I_{MT} \otimes 1_{L}) w
\end{array}
\end{align*}$$

where $y$ denotes the $MT$ column vector of observations of the dependent variables; $X$ denotes the $(MT \times KM)$ matrix of observations for the independent variables; $p$ denotes the $LKM$ column vector of weights associated with the points in $Z$; $w$ denotes the $LMT$ column vector of weights associated with the points in $V$; $1_L$, $1_{KM}$, and $1_{MT}$ are $L$, $KM$, and $MT$-dimensional column vectors, respectively, of ones; and $I_{KM}$ and $I_{MT}$ are $(KM \times KM)$ and $(MT \times MT)$ dimensional identity matrices. The subscript $l$ denotes the support point ($l = 1, 2, \ldots, L$), $k$ denotes the parameter ($k = 1, 2, \ldots, K$), $m$ denotes the equation ($m = 1, 2, \ldots, M$), and $t$ denotes the observation ($t = 1, 2, \ldots, T$).

Using this notation, the maximum entropy problem that is analogous to the OLS problem used as the initial step of the traditional SUR approach is
The results are GME-SUR estimates with independent errors, the analog of OLS. The covariance matrix $\hat{\Sigma}$ is computed based on the residual of the equations, $Vw = e$. An $L/L$ factorization of the $\hat{\Sigma}$ is used to compute the square root of the matrix.

After solving this problem, these entropy-based estimates are analogous to the Aitken two-step estimator. For iterative GME-SUR, the covariance matrix of the errors is recomputed, and a new $\hat{\Sigma}$ is computed and factored. As in traditional ITSUR, this process repeats until the covariance matrix and the parameter estimates converge.

The estimation of the parameters for the normed-moment version of SUR (GME-SUR-NM) uses an identical process. The constraints for GME-SUR-NM is defined as:

$$X'y = X'(S^{-1} \otimes I)XZp + X'(S^{-1} \otimes I)Vw$$

The estimation of the parameters for GME-SUR-NM uses an identical process as outlined previously for GME-SUR.

---

**Generalized Maximum Entropy for Multinomial Discrete Choice Models**

Multinomial discrete choice models take the form of an experiment that consists of $n$ trials. On each trial, one of $k$ alternatives is observed. If $y_{ij}$ is the random variable that takes on the value 1 when alternative $j$ is selected for the $i$th trial and 0 otherwise, then the probability that $y_{ij}$ is 1, conditional on a vector of regressors $X_i$ and unknown parameter vector $\beta_j$, is

$$\Pr(y_{ij} = 1 | X_i, \beta_j) = G(X'_i \beta_j)$$

where $G()$ is a link function. For noisy data the model becomes:

$$y_{ij} = G(X'_i \beta_j) + \epsilon_{ij} = p_{ij} + \epsilon_{ij}$$

The standard maximum likelihood approach for multinomial logit is equivalent to the maximum entropy solution for discrete choice models. The generalized maximum entropy approach avoids an assumption of the form of the link function $G()$.

The generalized maximum entropy for discrete choice models (GME-D) is written in primal form as

$$\text{maximize} \quad H(p, w) = -p' \ln(p) - w' \ln(w)$$
$$\text{subject to} \quad (I_j \otimes X'y) = (I_j \otimes X')p + (I_j \otimes X')Vw$$
$$\sum_j p_{ij} = 1 \quad \text{for } i = 1 \text{ to } N$$
$$\sum_m w_{ijm} = 1 \quad \text{for } i = 1 \text{ to } N \text{ and } j = 1 \text{ to } k$$

Golan, Judge, and Miller (1996) have shown that the dual unconstrained formulation of the GME-D can be viewed as a general class of logit models. Additionally, as the sample size increases, the solution of the dual problem approaches the maximum likelihood solution. Because of these characteristics, only the dual approach is available for the GME-D estimation method.

The parameters $\beta_j$ are the Lagrange multipliers of the constraints. The covariance matrix of the parameter estimates is computed as the inverse of the Hessian of the dual form of the objective function.
Censored or Truncated Dependent Variables

In practice, you might find that variables are not always measured throughout their natural ranges. A given variable might be recorded continuously in a range, but, outside of that range, only the endpoint is denoted. In other words, say that the data generating process is:

\[ y_i = x_{i*} + \epsilon. \]

However, you observe the following:

\[
y_i^* = \begin{cases} 
ub & : y_i \geq \ub \\
x_{i*} + \epsilon & : lb < y_i < ub \\
\lb & : y_i \leq \lb
\end{cases}
\]

The primal problem is simply a slight modification of the primal formulation for GME-GCE. You specify different supports for the errors in the truncated or censored region, perhaps reflecting some nonsample information. Then the data constraints are modified. The constraints that arise in the censored areas are changed to inequality constraints (Golan, Judge, and Perloff 1997). Let the variable \(X^u\) denote the observations of the explanatory variable where censoring occurs from the top, \(X^l\) from the bottom, and \(X^a\) in the middle region (no censoring). Let, \(V^u\) be the supports for the observations at the upper bound, \(V^l\) lower bound, and \(V^a\) in the middle.

You have:

\[
\begin{bmatrix}
y^u \geq \ub \\
y^a \\
y^l \leq \lb
\end{bmatrix} = \begin{bmatrix}
X^u \\
X^a \\
X^l
\end{bmatrix} Zp + \begin{bmatrix}
V^u w^u \\
V^a w^a \\
V^l w^l
\end{bmatrix}
\]

The primal problem then becomes

\[
\begin{align*}
\text{maximize} & \quad H(p, w) = -p' \ln(p) - w' \ln(w) \\
\text{subject to} & \quad y^a = X^a V^a p + V^a w^a \\
& \quad y^u \geq X^u V^u p + V^u w^u \\
& \quad y^l \leq X^l V^l p + V^l w^l \\
& \quad 1_K = (I_K \otimes 1_L') p \\
& \quad 1_T = (I_T \otimes 1_L') w
\end{align*}
\]

PROC ENTROPY requires that the number of supports be identical for all three regions.

Alternatively, you can think of cases where the dependent variable is observed continuously for most of its range. However, the variable’s range is reported for some observations. Such data is often found in highly disaggregated state level employment measures.

\[
y_i^* = \begin{cases} 
\text{missing} & : l_1 \leq y \leq r_1 \\
\vdots & : \vdots \\
\text{missing} & : l_k \leq y \leq r_k \\
x_{i*} + \epsilon & : \text{otherwise}
\end{cases}
\]

Just as in the censored case, each range yields two inequality constraints for each observation in that range.
Information Measures

PROC ENTROPY returns several measures of fit. First, the value of the objective function is returned. Next, the signal entropy is provided followed by the noise entropy. The sum of the noise and signal entropies should equal the value of the objective function. The next two metrics that follow are the normed entropies of both the signal and the noise.

Normalized entropy (NE) measures the relative informational content of both the signal and noise components through $p$ and $w$, respectively (Golan, Judge, and Miller 1996). Let $S$ denote the normalized entropy of the signal, $X\beta$, defined as:

$$S(\hat{p}) = \frac{-\hat{p}' \ln(\hat{p})}{-q' \ln(q)}$$

where $S(\hat{p}) \in [0, 1]$. In the case of GME, where uniform priors are assumed, $S$ can be written as:

$$S(\hat{p}) = \frac{-\hat{p}' \ln(\hat{p})}{\sum_i \ln(M_i)}$$

where $M_i$ is the number of support points for parameter $i$. A value of 0 for $S$ implies that there is no uncertainty regarding the parameters; hence, it is a degenerate situation. However, a value of 1 implies that the posterior distributions equal the priors, which indicates total uncertainty if the priors are uniform.

Because NE is relative, it can be used for comparing various situations. Consider adding a data point to the model. If $S_{T+1} = S_T$, then there is no additional information contained within that data constraint. However, if $S_{T+1} < S_T$, then the data point gives a more informed set of parameter estimates.

NE can be used for determining the importance of particular variables with regard to the reduction of the uncertainty they bring to the model. Each of the $k$ parameters that is estimated has an associated NE defined as

$$S(\hat{p}_k) = \frac{-\hat{p}'_k \ln(\hat{p}_k)}{-\ln(q_k)}$$

or, in the GME case,

$$S(\hat{p}_k) = \frac{-\hat{p}'_k \ln(\hat{p}_k)}{\ln(M)}$$

where $\hat{p}_k$ is the vector of supports for parameter $\beta_k$ and $M$ is the corresponding number of support points. Since a value of 1 implies no relative information for that particular sample, Golan, Judge, and Miller (1996) suggest an exclusion criteria of $S(\hat{p}_k) > 0.99$ as an acceptable means of selecting noninformative variables. See Golan, Judge, and Miller (1996) for some simulation results.

The final set of measures of fit are the parameter information index and error information index. These measures can be best summarized as $1 -$ the appropriate normed entropy.
### Parameter Covariance For GCE

For the cross-entropy problem, the estimate of the asymptotic variance of the signal parameter is given by:

\[
\hat{V}ar(\hat{\beta}) = \frac{\hat{\sigma}_{\beta}^2(\hat{\beta})}{\hat{\psi}^2(\hat{\beta})}(X'X)^{-1}
\]

where

\[
\hat{\sigma}_{\beta}^2(\hat{\beta}) = \frac{1}{N} \sum_{i=1}^{N} \gamma_i^2
\]

and \(\gamma_i\) is the Lagrange multiplier associated with the \(i\) th row of the \(Vw\) constraint matrix. Also,

\[
\hat{\psi}^2(\hat{\beta}) = \left[\frac{1}{N} \sum_{i=1}^{N} \left( \sum_{j=1}^{J} v_{ij}^2 w_{ij} - \left( \sum_{j=1}^{J} v_{ij} w_{ij}\right)\right)^2 \right]^{-1}
\]

### Parameter Covariance For GCE-M

Golan, Judge, and Miller (1996) give the finite approximation to the asymptotic variance matrix of the moment formulation as:

\[
\hat{V}ar(\hat{\beta}) = \Sigma_z X'X C^{-1} DC^{-1} X'X \Sigma_z
\]

where

\[
C = X'X \Sigma_z X'X + \Sigma_v
\]

and

\[
D = X'\Sigma_e X
\]

Recall that in the moment formulation, \(V\) is the support of \(\frac{X'X}{T}\), which implies that \(\Sigma_v\) is a \(k\)-dimensional variance matrix. \(\Sigma_z\) and \(\Sigma_v\) are both diagonal matrices with the form

\[
\Sigma_z = \begin{bmatrix}
\sum_{l=1}^{L} z_{1l}^2 p_{1l} - \left(\sum_{l=1}^{L} z_{1l} p_{1l}\right)^2 & 0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & \sum_{l=1}^{L} z_{Ll}^2 p_{Ll} - \left(\sum_{l=1}^{L} z_{Ll} p_{Ll}\right)^2
\end{bmatrix}
\]

and

\[
\Sigma_v = \begin{bmatrix}
\sum_{j=1}^{J} v_{1j}^2 w_{1j} - \left(\sum_{j=1}^{J} v_{1j} w_{1j}\right)^2 & 0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & \sum_{j=1}^{J} v_{Jj}^2 w_{Jj} - \left(\sum_{j=1}^{J} v_{Jj} w_{Jj}\right)^2
\end{bmatrix}
\]
Statistical Tests

Since the GME estimates have been shown to be asymptotically normally distributed, the classical Wald, Lagrange multiplier, and likelihood ratio statistics can be used for testing linear restrictions on the parameters.

Wald Tests

Let $H_0 : L\beta = m$, where $L$ is a set of linearly independent combinations of the elements of $\beta$. Then under the null hypothesis, the Wald test statistic,

$$T_W = (L\beta - m)' \left( L(V\hat{\text{a}}r(\hat{\beta}))L' \right)^{-1} (L\beta - m)$$

has a central $\chi^2$ limiting distribution with degrees of freedom equal to the rank of $L$.

Pseudo-Likelihood Ratio Tests

Using the conditionally maximized entropy function as a pseudo-likelihood, $F$, Mittelhammer and Cardell (2000) state that:

$$\frac{2\hat{\psi}(\hat{\beta})}{\sigma^2(\hat{\beta})} \left( F(\hat{\beta}) - F(\tilde{\beta}) \right)$$

has the limiting distribution of the Wald statistic when testing the same hypothesis. Note that $F(\hat{\beta})$ and $F(\tilde{\beta})$ are the maximum values of the entropy objective function over the full and restricted parameter spaces, respectively.

Lagrange Multiplier Tests

Again using the GME function as a pseudo-likelihood, Mittelhammer and Cardell (2000) define the Lagrange multiplier statistic as:

$$\frac{1}{\sigma^2(\hat{\beta})} G(\tilde{\beta})'(X'X)^{-1}G(\tilde{\beta})$$

where $G$ is the gradient of $F$, which is being evaluated at the optimum point for the restricted parameters. This test statistic shares the same limiting distribution as the Wald and pseudo-likelihood ratio tests.

Missing Values

If an observation in the input data set contains a missing value for any of the regressors or dependent values, that observation is dropped from the analysis.
Input Data Sets

DATA= Data Set

The DATA= data set specified in the PROC ENTROPY statement is the data set that contains the data to be analyzed.

PDATA= Data Set

The PDATA= data set specified in the PROC ENTROPY statement specifies the support points and prior probabilities to be used in the estimation. The PDATA= can be used in lieu of a PRIORS statement, but is intended for use in conjunction with the OUTP= option. Once priors are entered through a PRIORS statement, they can be reused in subsequent estimations by specifying the PDATA= option.

The variables in the data set are as follows:

- BY variables (if any)
- _TYPE_, a character variable of length 8 that identifies the estimation method: GME or GMEM. This is an optional column.
- variable, a character variable of length 32 that indicates the name of the regressor. The regressor name and the equation name identify a unique coefficient. This is required.
- _OBS_, a numeric variable that is either missing when the probabilities are for coefficients or the observation number when the probabilities are for the residual terms. The _OBS_ and the equation name identify which residual the probability is associated with. This an optional column.
- equation, a character variable of length 32 indicating the name of the dependent variable. This is a required column.
- NSupport, a numeric variable that indicates the number of support points for each basis. This variable is required.
- support, a numeric variable that is the support value the probability is associated with. This is a required column.
- prior, a numeric variable that is the prior probability associated with the probability. This is a required column.
- Prb, a numeric variable that is the estimated probability. This is optional.

SDATA= Data Set

The SDATA= data set specifies a data set that provides the covariance matrix of the equation errors. The matrix read from the SDATA= data set is used for the equation covariance matrix (S matrix) in the estimation. (The SDATA= S matrix is used to provide only the initial estimate of S for the methods that iterate the S matrix.)
Output Data Sets

OUT= Data Set

The OUT= data set specified in the PROC ENTROPY statement contains residuals of the dependent variables computed from the parameter estimates. The ID and BY variables are also added to this data set.

OUTEST= Data Set

The OUTEST= data set contains parameter estimates and, if requested via the COVOUT option, estimates of the covariance of the parameter estimates.

The variables in the data set are as follows:

- BY variables
- _NAME_, a character variable of length 32, blank for observations that contain parameter estimates or a parameter name for observations that contain covariances
- _TYPE_, a character variable of length 8 that identifies the estimation method: GME or GMEM
- the parameters estimated

If the COVOUT option is specified, an additional observation is written for each row of the estimate of the covariance matrix of parameter estimates, with the _NAME_ values containing the parameter names for the rows.

OUTP= Data Set

The OUTP= data set specified in the PROC ENTROPY statement contains the probabilities estimated for each support point, as well as the support points and prior probabilities used in the estimation.

The variables in the data set are as follows:

- BY variables (if any)
- _TYPE_, a character variable of length 8 that identifies the estimation method: GME or GMEM.
- variable, a character variable of length 32 that indicates the name of the regressor. The regressor name and the equation name identify a unique coefficient.
- _OBS_, a numeric variable that is either missing when the probabilities are for coefficients or the observation number when the probabilities are for the residual terms. The _OBS_ and the equation name identify which residual the probability is associated with.
- equation, a character variable of length 32 that indicates the name of the dependent variable
- NSupport, a numeric variable that indicates the number of support points for each basis
- support, a numeric variable that is the support value the probability is associated with
- prior, a numeric variable that is the prior probability associated with the probability
- Prb, a numeric variable that is the estimated probability
OUTL= Data Set

The OUTL= data set specified in the PROC ENTROPY statement contains the Lagrange multiplier values for the underlying maximum entropy problem.

The variables in the data set are as follows:

- BY variables
- equation, a character variable of length 32 that indicates the name of the dependent variable
- variable, a character variable of length 32 that indicates the name of the regressor. The regressor name and the equation name identify a unique coefficient.
- _OBS_, a numeric variable that is either missing when the probabilities are for coefficients or the observation number when the probabilities are for the residual terms. The _OBS_ and the equation name identify which residual the Lagrange multiplier is associated with
- LagrangeMult, a numeric variable that contains the Lagrange multipliers

ODS Table Names

PROC ENTROPY assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table.

<table>
<thead>
<tr>
<th>ODS Table Name</th>
<th>Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvCrit</td>
<td>Convergence criteria for estimation</td>
<td>default</td>
</tr>
<tr>
<td>ConvergenceStatus</td>
<td>Convergence status</td>
<td>default</td>
</tr>
<tr>
<td>DatasetOptions</td>
<td>Data sets used</td>
<td>default</td>
</tr>
<tr>
<td>MinSummary</td>
<td>Number of parameters, estimation kind</td>
<td>default</td>
</tr>
<tr>
<td>ObsUsed</td>
<td>Observations read, used, and missing</td>
<td>default</td>
</tr>
<tr>
<td>ParameterEstimates</td>
<td>Parameter estimates</td>
<td>default</td>
</tr>
<tr>
<td>ResidSummary</td>
<td>Summary of the SSE, MSE for the equations</td>
<td>default</td>
</tr>
<tr>
<td>TestResults</td>
<td>Test statement table</td>
<td>TEST statement</td>
</tr>
</tbody>
</table>

ODS Graphics

Before you create graphs, ODS Graphics must be enabled (for example, with the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section “Enabling and Disabling ODS Graphics” in that chapter.

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section “A Primer on ODS Statistical Graphics” in that chapter.

This section describes the use of ODS for creating graphics with the ENTROPY procedure.

**ODS Graph Names**

PROC ENTROPY assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 14.3.

To request these graphs, you must specify the ODS GRAPHICS statement.

<table>
<thead>
<tr>
<th>ODS Graph Name</th>
<th>Plot Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiagnosticsPanel</td>
<td>Includes all the plots listed below</td>
</tr>
<tr>
<td>FitPlot</td>
<td>Predicted versus actual plot</td>
</tr>
<tr>
<td>CooksD</td>
<td>Cook’s D plot</td>
</tr>
<tr>
<td>QQPlot</td>
<td>Q-Q plot of residuals</td>
</tr>
<tr>
<td>StudentResidualPlot</td>
<td>Studentized residual plot</td>
</tr>
<tr>
<td>ResidualHistogram</td>
<td>Histogram of the residuals</td>
</tr>
</tbody>
</table>

**Examples: ENTROPY Procedure**

**Example 14.1: Nonnormal Error Estimation**

This example illustrates the difference between GME-M and GME. One of the basic assumptions of OLS estimation is that the errors in the estimation are normally distributed. If this assumption is violated, the estimated parameters are biased. For GME-M, the story is similar. If the first moment of the distribution of the errors and a scale factor cannot be used to describe the distribution, then the parameter estimates from GME-MN are more biased. GME is much less sensitive to the underlying distribution of the errors than GME-M.

To illustrate this, data for the following model is simulated with three different error distributions:

\[ y = a \cdot x_1 + b \cdot x_2 + \epsilon. \]

For the first simulation, \( \epsilon \) is distributed normally, then a chi-squared distribution with six degrees of freedom is assumed for the second simulation, and finally \( \epsilon \) is assumed to have a Cauchy distribution in the third simulation.

In each of the three simulations, 100 samples of 10 observations each were simulated. The data for the model with the Cauchy error distribution is generated using the following DATA step code:
data one;
call streaminit(156789);
do by = 1 to 100;
do x2 = 1 to 10;
x1 = 10 * ranuni( 512);
y = x1 + 2*x2 + rand('cauchy');
output;
end;
end;
run;
The statements for the other distributions are identical except for the argument to the RAND() function.
The parameters to the model were estimated by using maximum entropy with the following programming statements:

```
proc entropy data=one gme outest=parm1;
    model y = x1 x2;
    by by;
run;
```
The estimation by using moment-constrained maximum entropy was performed by changing the GME option to GMEM. For comparison, the same model was estimated by using OLS with the following PROC REG statements:

```
proc reg data=one outest=parm3;
    model y = x1 x2;
    by by;
run;
```
The 100 estimations of the coefficient on variable x1 are then summarized for each of the three error distributions by using PROC UNIVARIATE, as follows:

```
proc univariate data=parm1;
    var x1;
run;
```
The following table summarizes the results from the estimations. The true value for the coefficient on x1 is 1.0.

<table>
<thead>
<tr>
<th>Estimation Method</th>
<th>Normal Mean</th>
<th>Normal Std Deviation</th>
<th>Chi-Squared Mean</th>
<th>Chi-Squared Std Deviation</th>
<th>Cauchy Mean</th>
<th>Cauchy Std Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GME</td>
<td>0.418</td>
<td>0.117</td>
<td>0.626</td>
<td>0.330</td>
<td>0.818</td>
<td>3.36</td>
</tr>
<tr>
<td>GME-M</td>
<td>0.878</td>
<td>0.116</td>
<td>0.948</td>
<td>0.427</td>
<td>3.03</td>
<td>13.62</td>
</tr>
<tr>
<td>OLS</td>
<td>0.973</td>
<td>0.142</td>
<td>1.023</td>
<td>0.467</td>
<td>5.54</td>
<td>26.83</td>
</tr>
</tbody>
</table>

For normally distributed or nearly normally distributed data, moment-constrained maximum entropy is a good choice. For distributions not well described by a normal distribution, data-constrained maximum entropy is a good choice.
Example 14.2: Unreplicated Factorial Experiments

Factorial experiments are useful for studying the effects of various factors on a response. For the practitioner constrained to the use of OLS regression, there must be replication to estimate all of the possible main and interaction effects in a factorial experiment. Using OLS regression to analyze unreplicated experimental data results in zero degrees of freedom for error in the ANOVA table, since there are as many parameters as observations. This situation leaves the experimenter unable to compute confidence intervals or perform hypothesis testing on the parameter estimates.

Several options are available when replication is impossible. The higher-order interactions can be assumed to have negligible effects, and their degrees of freedom can be pooled to create the error degrees of freedom used to perform inference on the lower-order estimates. Or, if a preliminary experiment is being run, a normal probability plot of all effects can provide insight as to which effects are significant, and therefore focused, in a later, more complete experiment.

The following example illustrates the probability plot methodology and the alternative by using PROC ENTROPY. Consider a $2^4$ factorial model with no replication. The data are taken from Myers and Montgomery (1995).

```plaintext
data rate;
  do a=-1,1; do b=-1,1; do c=-1,1; do d=-1,1;
    input y @@;
    ab=a*b; ac=a*c; ad=a*d; bc=b*c; bd=b*d; cd=c*d;
    abc=a*b*c; abd=a*b*d; acd=a*c*d; bcd=b*c*d;
    abcd=a*b*c*d;
    output;
  end; end; end; end;
  datalines;
  45 71 48 65 68 60 80 65 43 100 45 104 75 86 70 96
  ;
run;
```

Analyze the data by using PROC REG, then output the resulting estimates.

```plaintext
proc reg data=rate outest=regout;
  model y=a b c d ab ac ad bc bd cd abc abd acd bcd abcd;
run;

proc transpose data=regout out=ploteff name=effect prefix=est;
  var a b c d ab ac ad bc bd cd abc abd acd bcd abcd;
run;
```

Now the normal scores for the estimates can be computed with the rank procedure as follows:

```plaintext
proc rank data=ploteff normal=blom out=qqplot;
  var est1;
  ranks normalq;
run;
```
To create the probability plot, simply plot the estimates versus their normal scores by using PROC SGPLOT as follows:

```sas
title "Unreplicated Factorial Experiments";
proc sgplot data=qqplot;
   scatter x=est1 y=normalq / markerchar=effect
       markercharattrs=(size=10pt);
   xaxis label="Estimate";
   yaxis label="Normal Quantile";
run;
```

*Output 14.2.1* Normal Probability Plot of Effects

The plot shown in *Output 14.2.1* displays evidence that the \(a\), \(b\), \(d\), \(ad\), and \(bd\) estimates do not fit into the purely random normal model, which suggests that they may have some significant effect on the response variable. To verify this, fit a reduced model that contains only these effects.
Example 14.2: Unreplicated Factorial Experiments

The estimates for the reduced model are shown in Output 14.2.2.

**Output 14.2.2 Reduced Model OLS Estimates**

| Parameter      | Estimate | Standard Error | t Value | Pr > |t| |
|----------------|----------|----------------|---------|------|---|
| Intercept      | 70.06250 | 1.10432        | 63.44   | <.0001 |
| a              | 7.31250  | 1.10432        | 6.62    | <.0001 |
| b              | 4.93750  | 1.10432        | 4.47    | 0.0012 |
| d              | 10.81250 | 1.10432        | 9.79    | <.0001 |
| ad             | 8.31250  | 1.10432        | 7.53    | <.0001 |
| bd             | -9.06250 | 1.10432        | -8.21   | <.0001 |

These results support the probability plot methodology.

PROC ENTROPY can directly estimate the full model without having to rely upon the probability plot for insight into which effects can be significant. To illustrate this, PROC ENTROPY is run by using default parameter and error supports in the following statements:

```
proc entropy data=rate;
    model y=a b c d ab ac ad bc bd cd abc abd acd bcd abcd;
run;
```

The resulting GME estimates are shown in Output 14.2.3. Note that the parameter estimates associated with the a, b, d, ad, and bd effects are all significant.
Example 14.3: Censored Data Models in PROC ENTROPY

Data available to an analyst might sometimes be censored, where only part of the actual series is observed. Consider the case in which only observations greater than some lower bound are recorded, as defined by the following process:

\[ y = \max(X\beta + \epsilon, lb) . \]

Running ordinary least squares estimation on data generated by the preceding process is not optimal because the estimates are likely to be biased and inefficient. One alternative to estimating models with censored data is the tobit estimator. This model is supported in the QLIM procedure in SAS/ETS and in the LIFEREG procedure in SAS/STAT. PROC ENTROPY provides another alternative which can make it very easy to estimate such a model correctly.

The following DATA step generates censored data in which any negative values of the dependent variable, \( y \), are set to a lower bound of 0.
data cens;
   do t = 1 to 100;
      x1 = 5 * ranuni(456);
      x2 = 10 * ranuni(456);
      y = 4.5*x1 + 2*x2 + 15 * rannor(456);
      if( y<0 ) then y = 0;
      output;
   end;
run;

To illustrate the effect of the censored option in PROC ENTROPY, the model is initially estimated without accounting for censoring in the following statements:

```
   title "Censored Data Estimation";
   proc entropy data = cens gme primal;
      priors intercept -32 32
                      x1    -15 15
                      x2    -15 15;
      model y = x1 x2 /
                 esupports = (-25 1 25);
   run;
```

```
Output 14.3.1  GME Estimates

Censored Data Estimation

The ENTROPY Procedure

GME Variable   Estimate Mean Std Error   t Value Pr > |t|
   x1           2.377609 0.000503 4725.98 <.0001
   x2           2.353014 0.000255 9244.87 <.0001
intercept     5.478121 0.00188 2906.41 <.0001
```

The previous model is reestimated by using the CENSORED option in the following statements:

```
   proc entropy data = cens gme primal;
      priors intercept -32 32
                      x1    -15 15
                      x2    -15 15;
      model y = x1 x2 /
                 esupports = (-25 1 25)
                 censored(lb = 0, esupports=(-15 1 15));
   run;
```
The second set of entropy estimates are much closer to the true parameter estimates of 4.5 and 2. Since another alternative available for fitting a model of censored data is a tobit model, PROC QLIM is used in the following statements to fit a tobit model to the data:

```
proc qlim data=cens;
  model y = x1 x2;
  endogenous y ~ censored(lb=0);
run;
```

Output 14.3.2 Entropy Estimates

Censored Data Estimation

The ENTROPY Procedure

| Variable | Estimate | Approx Std Err | Approx t Value | Approx Pr > |t|
|----------|----------|----------------|----------------|-------------|
| x1       | 4.429697 | 0.00690        | 641.85         | <.0001      |
| x2       | 1.46858  | 0.00349        | 420.61         | <.0001      |
| intercept| 8.261412 | 0.0259         | 319.51         | <.0001      |

For this data and code, PROC ENTROPY produces estimates that are closer to the true parameter values than those computed by PROC QLIM.

Example 14.4: Use of the PDATA= Option

It is sometimes useful to specify priors and supports by using the PDATA= option. This example illustrates how to create a PDATA= data set which contains the priors and support points for use in a subsequent PROC ENTROPY step. In order to have a model to estimate in PROC ENTROPY, you must first have data to analyze. The following DATA step generates the data used in this analysis:
title "Using a PDATA= data set";
data a;
array x[4];
do t = 1 to 100;
  ys = -5;
  do k = 1 to 4;
    x[k] = rannor( 55372 ) ;
    ys = ys + x[k] * k;
  end;
  ys = ys + rannor( 55372 );
output;
end;
run;

Next you fit this data with some arbitrary parameter support points and priors by using the following PROC ENTROPY statements:

   proc entropy data = a gme primal;
   priors x1 -10(2) 30(1)
           x2 -20(3) 30(2)
           x3 -15(4) 30(4)
           x4 -25(3) 30(2)
           intercept -13(4) 30(2) ;
   model ys = x1 x2 x3 x4 / esupports=(-25 0 25);
run;

These statements produce the output shown in Output 14.4.1.

| GME Variable | Estimate | Approx Std Err | t Value | Approx Pr > |t| |
|--------------|----------|----------------|---------|-------------|---|
| x1           | 1.195668 | 0.1078         | 11.09   | <.0001      |   |
| x2           | 1.844903 | 0.1018         | 18.12   | <.0001      |   |
| x3           | 3.268396 | 0.1136         | 28.77   | <.0001      |   |
| x4           | 3.908194 | 0.0934         | 41.83   | <.0001      |   |
| intercept    | -4.94319 | 0.1005         | -49.21  | <.0001      |   |

You can estimate the same model by first creating a PDATA= data set, which includes the same information as the PRIORS statement in the preceding PROC ENTROPY step.
A data set that defines the supports and priors for the model parameters is shown in the following statements:

```plaintext
data test;
  length Variable $ 12 Equation $ 12;
  input Variable $ Equation $ Nsupport Support Prior ;
datalines;
  Intercept . 2 -13 0.66667
  Intercept . 2 30 0.33333
  x1 . 2 -10 0.66667
  x1 . 2 30 0.33333
  x2 . 2 -20 0.60000
  x2 . 2 30 0.40000
  x3 . 2 -15 0.50000
  x3 . 2 30 0.50000
  x4 . 2 -25 0.60000
  x4 . 2 30 0.40000
;
```

The following statements reestimate the model by using these support points.

```plaintext
proc entropy data=a gme primal pdata=test;
  model ys = x1 x2 x3 x4 / esupports=(-25 0 25);
run;
```

These statements produce the output shown in Output 14.4.2.

**Output 14.4.2** Output From PROC ENTROPY with PDATA= option

Using a PDATA= data set

The ENTROPY Procedure

| Variable | Estimate  | Approx Std Err | t Value | Approx Pr > |t| |
|----------|-----------|----------------|---------|-------------|---|
| x1       | 1.95686   | 0.1078         | 11.09   | <.0001      |
| x2       | 1.844902  | 0.1018         | 18.12   | <.0001      |
| x3       | 3.268395  | 0.1136         | 28.77   | <.0001      |
| x4       | 3.908194  | 0.0934         | 41.83   | <.0001      |
| Intercept| -4.94319  | 0.1005         | -49.21  | <.0001      |

These results are identical to the ones produced by the previous PROC ENTROPY step.

**Example 14.5: Illustration of ODS Graphics**

This example illustrates how to use ODS graphics in the ENTROPY procedure. This example is a continuation of the example in the section “Simple Regression Analysis” on page 788. Graphical displays are requested by specifying the ODS GRAPHICS statement. For information about the graphics available in the ENTROPY procedure, see the section “ODS Graphics” on page 832.
The following statements show how to generate ODS graphics plots with the ENTROPY procedure. The plots are displayed in Output 14.5.1.

```
proc entropy data=coleman;
  model test_score = teach_sal prcnt_prof socio_stat
                  teach_score mom_ed;
run;
```

**Output 14.5.1 Model Diagnostics Plots**

![Image of Model Diagnostics Plots]

**Observations 18  MSE 8.575791  Model DF 6**
References


bounds on parameter estimates, 813
BOUNDPS statement, 813

ENTROPY procedure
  input data sets, 830
  missing values, 829
  ODS graph names, 833
  output data sets, 831
  output table names, 832

equality restriction
  linear models, 817

inequality restriction
  linear models, 817

input data sets
  ENTROPY procedure, 830

Lagrange multiplier test
  linear hypotheses, 818

likelihood ratio test
  linear hypotheses, 818

linear hypotheses
  Lagrange multiplier test, 818
  likelihood ratio test, 818
  Wald test, 818

linear models
  equality restriction, 817
  inequality restriction, 817
  restricted estimation, 817

missing values
  ENTROPY procedure, 829

ODS graph names
  ENTROPY procedure, 833
output data sets
  ENTROPY procedure, 831
output table names
  ENTROPY procedure, 832

RESTRICT statement, 817
restricted estimation
  linear models, 817

tests of parameters, 817

Wald test
  linear hypotheses, 818
Syntax Index

ALL option
   TEST statement (ENTROPY), 818

BOUNDS statement
   ENTROPY procedure, 813

BY statement
   ENTROPY procedure, 815

CENSORED option
   MODEL statement (ENTROPY), 816

COLLIN option
   ENTROPY procedure, 810

CONVERGE= option
   ENTROPY procedure, 812

COVBEST= option
   ENTROPY procedure, 810

COVOUT option
   ENTROPY procedure, 811

DATA= option
   ENTROPY procedure, 810

DUAL option
   ENTROPY procedure, 812

ENTROPY procedure, 808
   syntax, 808

ESUPPORTS= option
   MODEL statement (ENTROPY), 815

GCE option
   ENTROPY procedure, 810

GCEM option
   ENTROPY procedure, 810

GCONV= option
   ENTROPY procedure, 812

GME option
   ENTROPY procedure, 810

GMED option
   ENTROPY procedure, 810

GMEM option
   ENTROPY procedure, 810

ID statement
   ENTROPY procedure, 815

ITPRINT option
   ENTROPY procedure, 811

LAGRANGE option
   TEST statement (ENTROPY), 818

LIKE option
   TEST statement (ENTROPY), 818

LM option
   TEST statement (ENTROPY), 818

LR option
   TEST statement (ENTROPY), 818

MARGINALS option
   MODEL statement (ENTROPY), 816

MARKOV option
   ENTROPY procedure, 810

MAXITER= option
   ENTROPY procedure, 812

MAXSUBITER= option
   ENTROPY procedure, 812

METHOD= option
   ENTROPY procedure, 812

MODEL statement
   ENTROPY procedure, 815

NOINT option
   MODEL statement (ENTROPY), 816

NOPRINT option
   PROC ENTROPY statement, 811

OUT= option
   ENTROPY procedure, 811
   TEST statement (ENTROPY), 818

OUTCOV option
   ENTROPY procedure, 811

OUTTEST= option
   ENTROPY procedure, 811
   ENTROPY statement, 831

OUTL= option
   ENTROPY procedure, 811
   ENTROPY statement, 832

OUTP= option
   ENTROPY procedure, 811
   ENTROPY statement, 831

OUTS= option
   ENTROPY procedure, 811

OUTSUSED= option
   ENTROPY procedure, 811

PDATA= option
   ENTROPY procedure, 810
   ENTROPY statement, 830

PLOTS option
   PROC ENTROPY statement, 811
PRIMAL option
   ENTROPY procedure, 812
PRIORS statement
   ENTROPY procedure, 816
PROC ENTROPY statement, 810
PURE option
   ENTROPY procedure, 810

RANGE option
   MODEL statement (ENTROPY), 816
RAO option
   TEST statement (ENTROPY), 818
RESTRICT statement
   ENTROPY procedure, 817

SDATA= option
   ENTROPY procedure, 811, 830
SUR option
   ENTROPY procedure, 810

TECH= option
   ENTROPY procedure, 812
TECHNIQUE= option
   ENTROPY procedure, 812
TEST statement
   ENTROPY procedure, 817

VARDEF= option
   Proc ENTROPY, 810

WALD option
   TEST statement (ENTROPY), 818
WEIGHT statement
   ENTROPY procedure, 819