

Chapter 487

Hybrid Appraisal Models (Old Version)

Introduction

This procedure analyzes a special prediction model often used in mass appraisal and assessment. The model is referred to as the *hybrid* model. Although NCSS has a nonlinear regression module for solving this model, this module was added for several reasons:

1. It provides several methods of optimization for calibrating the model parameters. Some of these methods have the same goal as the ‘feedback algorithm’.
2. It streamlines model setup, including the automatic generation of binary variables.
3. It provides many new reports and statistics that aid in the calibration of the parameter values.

The use of multiple and nonlinear regression in property appraisal and assessment administration has been encouraged by such organizations as the International Association of Assessing Officers (IAAO). They publish a book by Eckert (1990) and teach courses which use these regression procedures. This program was developed to automate the hybrid model that they propose.

The Hybrid Appraisal Model

The hybrid model is a combination of both additive and multiplicative models. It relates the sales price of a property to various characteristics such as size (in square feet), lot size, construction quality, location, number of bathrooms, etc. This model computes a market value for each ‘structure’ of the parcel, where a structure refers to an object like the land or a build. For example, the sales price may be the sum of the market values of the building, the lot, and a garage. Because of variance that may occur because of location, time, etc., this sum may be adjusted by one or more overall variables. These overall variables act as percentage adjustments.

The general form of the model is:

$$\text{Sales Price} = \text{Overall} (\text{Building} + \text{Land} + \text{Garage} + \dots)$$

Each of these factors are modeled by one or more variables from the database. These factor models are made up of three types of variables: rate, binary, and amount. The amount variable usually represents the ‘size’ of the structure. Examples are acreage and square footage. The rate and binary variables are variables that adjust the size variables up or down, such as a quality index or an age adjustment.

Hybrid Appraisal Models (Old Version)

Using this construction, the individual factors are modeled as follows.

$$\text{Overall} = R_1^{B_1} R_2^{B_2} B_3^{I_3} B_4^{I_4}$$

$$\text{Building} = R_5^{B_5} R_6^{B_6} B_7^{I_7} B_8^{I_8} (B_9 A_9 + B_{10} A_{10})$$

$$\text{Land} = R_{11}^{B_{11}} R_{12}^{B_{12}} B_{13}^{I_{13}} B_{14}^{I_{14}} (B_{15} A_{15} + B_{16} A_{16})$$

$$\text{Garage} = R_{17}^{B_{17}} R_{18}^{B_{18}} B_{19}^{I_{19}} B_{20}^{I_{20}} (B_{21} A_{21} + B_{22} A_{22})$$

In this model, R_i represent rates or multipliers (usually centered near one) which modify the whole factor. Examples of rate variables include sales date multiplier and depreciation. These are analogous to percentage adjustment variables.

The I_i 's represent indicator (binary) variables. These are variables that have only two values: zero and one. Examples of these are neighborhood indicators and special feature indicators (such presence of a swimming pool). Usually, qualitative variables are broken down into individual binary variables. For example, suppose that the properties in your study come from three neighborhoods. You would designate one as the standard neighborhood, then create two binary variables, one for each of the other neighborhoods. The model will then adjust for differences among the three neighborhoods.

The A_i 's represent amount variables like lot size or square footage of living area. These variables are entered in the normal linear fashion.

The B_i 's represent the coefficients that are estimated from the data. Often, these coefficients are constrained to lie within specified limits.

Note that the number of variables in each group varies. Quite often you may have ten or fifteen binary variables and only one rate variable.

Differential Evolution

Introduction

One of the steps in using the hybrid model in mass appraisal is to calibrate (estimate) the unknown coefficient values, the B 's. Often, the method of least squares (MRA) is used to estimate the coefficients. Least squares finds the set of estimates that minimize the sum of the squared errors. That is, the objective of the method of least squares is to minimum the sum of squared errors. The sum of squared errors is called the *objective function*, and the problem is to minimize it. Least squares is one method of minimizing it. Another possibility is simple trial-and-error (of course, trial-and-error may be very time consuming).

Because of the distortion that a few anomalies in the data can cause when least squares is used, other methods have been proposed. One alternative is the *feedback* algorithm. This algorithm seeks to minimize a different objective function: the *average absolute percent error*. This objective function quantifies the percentage accuracy of the model. Both expensive and inexpensive parcels are modeled to the same percentage accuracy. Least squares, on the other hand, concentrates on fitting the most expensive properties. The feedback algorithm is used to minimize an specific objective function: the *average absolute percent error*. Recently, mathematicians have found other algorithms with the same goal of minimizing an objective function. One such algorithm is *differential evolution*.

Differential evolution is one of a group of *genetic algorithms* (see for example, the recent book by Haupt (1998)). By studying how generations respond over time to their environment, mathematicians have discovered new, more robust, algorithms for minimizing an objective function. Differential evolution is one of these algorithms. It can be outlined as follows. A *population* of about 20 individuals has certain traits (values of the unknown coefficients being estimated). The well-being of an individual is measured by his/her value of the objective function. Each individual gives birth to a new individual and then dies, thus forming a new generation. Each new individual

Hybrid Appraisal Models (Old Version)

inherits traits from their parents. The well-being of each child is computed, and if it is better than their parent, they take the parents place. Otherwise, the parent's traits pass directly to the child. Finally, occasionally, a mutation of a trait will occur.

The main point to remember is that the big goal is to minimize the objective function. Whether we use the feedback algorithm or the differential evolution algorithm makes little difference, as long as you find the minimum!

The Algorithm

A population consists of a small group (about 20) of individuals whose characteristics are the values of the unknown coefficients. Using these coefficients, the value of the objective function is computed for each individual. This value is an inverse measure of the wellbeing of an individual. The lower the value, the higher the well being.

To begin the algorithm, a small group of individuals must be formed. This is done by assigning the nonlinear regression coefficients to one individual and then randomly assigning the other individuals to a grid of values around this first individual. This is the initial population.

The next step is the evolution of the population. The population progresses through a series of *generations*. At each change in generation, depending on a member's wellbeing, each population member may move on to the next generation or be replaced by a better member. For each member, a trial replacement is constructed as follows. The best member of the population is found. The attributes of each replacement member are computed as a weighted average of those of the member and the best member. The amount of weight of the best member is controlled by the *inheritance factor*. This is a value between 0 and 1. The closer this value is to 1, the more the replacement member resembles the best member. The closer this value is to 0, the more the replacement member resembles their parent. The value of 0.85 seems to work in many cases.

As in real populations, *mutations* occur at a given rate. When a mutation occurs, a particular trait is changed randomly. This tends to maintain diversity in the population. A mutation rate of about 30% (0.30) seems to work well.

The algorithm proceeds from generation to generation until the population seems to converge to a single individual. The number of generations is arbitrary. Usually, about 100 generations are needed for the algorithm to converge.

Assumptions

The main assumption needed is that the data are well represented by the model.

Data Structure

The data are entered as one dependent variable and one or more independent variables. Each row of data represents a single parcel.

Missing Values

Rows with missing values in the variables being analyzed are ignored in the calculations. When only the value of the dependent variable is missing, predicted values are generated.

Procedure Options

This section describes the options available in this procedure.

Variables Tab

This panel specifies the variables used in the analysis.

Estimation Specification

Estimation Method

This option specifies the method used to estimate the model coefficients. Several methods are available.

- **Minimize Squared Errors (Nonlinear Regression)**

This is the classical approach often used by statisticians because it gives reasonable estimates relatively quickly. This method tends to emphasize relatively expensive properties as compared to less expensive properties.

- **Minimize the Average [Percent Error]**

Using the genetic search algorithm called *differential evolution*, this method finds estimates that minimize the average of the absolute percent errors. These percent errors are the difference between the actual and predicted sales prices divided by the actual price. This method treats all properties equally, irregardless of price.

The function minimized is

$$MAPE = \frac{\sum_{properties} \left| \frac{100(actual - predicted)}{actual} \right|}{N}$$

where $|X|$ represents the absolute (positive) value of X and $\sum_{properties} X$ is interpreted as the sum of the values of all properties. Note that this is the quantity minimized by the *feedback* algorithm.

If you want to use a solution that minimizes the percent errors, this is the solution that we recommend.

- **Minimize the Maximum [Percent Error]**

Using the genetic search algorithm called *differential evolution*, this method finds estimates that minimize the maximum of the absolute percent errors. These percent errors are the difference between the actual and predicted sales prices divided by the actual price.

- **Minimize the Median [Percent Error]**

Using the genetic search algorithm called *differential evolution*, this method finds estimates that minimize the median of the absolute percent errors. These percent errors are the difference between the actual and predicted sales prices divided by the actual price.

- **Minimize the Percentile [Percent Error]**

Using the genetic search algorithm called *differential evolution*, this method finds estimates that minimize a designated percentile of the absolute percent errors. These percent errors are the difference between the actual and predicted sales prices divided by the actual price. The percentile is specified in the Min Percentile box.

Hybrid Appraisal Models (Old Version)

- **Minimize the Average |Error|**

Using the genetic search algorithm called *differential evolution*, this method finds estimates that minimize the average of the absolute errors. These absolute errors are the difference between the actual and predicted sales prices. These percent errors are the difference between the actual and predicted sales prices divided by the actual price. The percentile is specified in the Min Percentile box.

- **Minimize the Median |Error|**

Using the genetic search algorithm called *differential evolution*, this method finds estimates that minimize the median of the absolute errors. These errors are the difference between the actual and predicted sales prices.

- **Minimize the Percentile |Error|**

Using the genetic search algorithm called *differential evolution*, this method finds estimates that minimize a designated percentile of the absolute errors. These errors are the difference between the actual and predicted sales prices. The percentile is specified in the Min Percentile box.

Min Percentile

This option specifies the percentile minimized when the Estimation Method is set to minimize a percentile. Although a value between 1 and 99 is legal, the routine should use a value between 50 and 95.

Dependent (Sales Price) Variable

Y - Dependent Variable (Sales Price)

This option specifies the sales price variable. This is the value that will be estimated by the hybrid model.

If you want to estimate the sales price of other properties, just add their values to the end of the database, but leave the sales price blank.

Note that the prices may be in any scale you want—dollars, hundreds of dollars, or thousands of dollars. When the resulting predicted sales prices are displayed, they will be in the same scale as these values.

Variable Specification

Factor

Specify the factor (object) that this parameter belongs to. Possible choices of factor are overall, building (up to 10), land, shed, garage, or pool.

Variable

Specify one or more variables with the same factor, type, minimum, maximum, and starting values. Usually, you will specify only one variable per line, but more are allowed if you desire.

- **Binaries**

If this is a type D (discrete) variable, you can specify a reference value in parentheses after the variable name. If you do not specify a reference value, the program sorts the values and picks the last value as the reference value. The reference value is that value for which no binary variable is generated. The number of binary (0-1) variables generated is always one less than the number of unique values.

For example, suppose you will use a variable called `ExtType` that has three possible values: B for brick, U for stucco, or S for siding. Further suppose that in your area, siding is the most common exterior type. Hence, siding would be the most obvious choice for the reference value. You would enter `ExtType(S)` for this variable. The program would generate two binary variables: one for brick and the other for stucco.

Hybrid Appraisal Models (Old Version)

- **Single Binary**

It is possible to specify that only a single binary be generated for a type D (discrete) variable. This is done by adding a comma and an 'I' after the reference value. When you do this, only a single binary variable is generated for the value indicated.

For example, using the exterior type example given above, the statement $ExtType(S,I)$ would cause the program to generate a single indicator variable that is '1' when the value of $ExtType$ is S and '0' otherwise.

Type

This option specifies the variable type. Possible choices are: (A)mount, (D)iscrete, and (R)ate.

- **(A)mount**

Amount variables represent the factor size. Examples are square footage and lot size. These enter into the prediction model as linear variables such as $B_{21}X_{21} + B_{22}X_{22}$, where the B 's are the estimated coefficients and the X 's are the variables.

- **(D)iscrete**

A discrete variable is one taking on only a view unique values, such as exterior type or neighborhood. In fact, discrete variables are not necessarily numeric. A set of indicator (binary) variables is generated for a discrete variable. This set of binary variables enters into the prediction model as $B_7^{X_7} B_8^{X_8}$, where the B 's are the estimated coefficients and the X 's are the binary variables. Note that since $B^0 = 1$ for any B , these variables make an adjustment when the binary variable is true (1), but have no effect when the binary variable is false (0)

- **(R)ate**

A rate variable is a multiplier (usually centered near one) which modifies the whole factor. Examples of rate variables include sales date multiplier and depreciation. These are analogous to percentage adjustment variables. They enter into the prediction model as multipliers using the construction $X_1^{B_1} X_2^{B_2}$, where the B 's are the estimated coefficients and the X 's are the rates.

Min Start Max

Enter the word 'Defaults' or an ascending set of three numbers separated by blanks or commas. If you enter the word 'Defaults', the corresponding default values as entered under the Defaults tab will be used.

If you enter a set of numbers, the first number is the minimum, the second is the starting value, and the third is the maximum value that the coefficient associated with the variable can take on. For example, the triplet '0.1, 1.0, 2.0' sets the minimum at 0.1, the maximum at 2.0, and the starting value at 1.0. The minimum and maximum allow you to define a range of possible values (hopefully, a realistic range) in which the search for the final estimate is to take place.

Minimum

This is the smallest value that the parameter can take on. The algorithm searches for a value between this and the Maximum. If you want to search in an unlimited range, enter a large negative number such as -1E9, which is -1,000,000,000. You can enter a -B (for negative one billion) if you want to leave the value free to vary.

Starting Value

This is the beginning value of the parameter. The algorithm searches for a value between the Minimum and the Maximum, beginning with this number. The closer this value is to the final value, the quicker the algorithm will converge.

Following are suggestions for selecting starting values.

1. Make sure that the starting values you supply are legitimate.

Hybrid Appraisal Models (Old Version)

2. Before you go to a lot of effort, make a few trial runs using starting values of 0.0, 0.5, and 1.0. Often, one of these values will converge.

Maximum

This is the largest value that the parameter can take on. The algorithm searches for a value between the Minimum and this value, beginning at the Starting Value. If you want to search in an unlimited range, enter a large positive number such as **1E9**, which is 1,000,000,000. You can enter a B (for positive one billion) if you want to leave the value free to vary.

Options Tab

Nonlinear Regression Options

This set of options controls the nonlinear regression algorithm. Note that when you are estimating using differential evolution, nonlinear regression is used to find appropriate starting values. Hence, these nonlinear regression options are always used.

Lambda

This is the starting value of the lambda parameter as defined in Marquardt's procedure. We recommend that you do not change this value unless you are very familiar with both your model and the Marquardt nonlinear regression procedure. Changing this value will influence the speed at which the algorithm converges.

Lambda Inc

This is a factor used for increasing lambda when necessary. It influences the rate at which the algorithm converges.

Max Iterations

This sets the maximum number of iterations before the nonlinear regression algorithm is aborted. If the starting values you have supplied are not appropriate or the model does not fit the data, the algorithm may diverge. Setting this value to an appropriate number (say 50) causes the algorithm to abort after this many iterations.

Nash Phi

Nash supplies a factor he calls *phi* for modifying lambda. When the residual sum of squares is large, increasing this value may speed convergence.

Lambda Dec

This is a factor used for decreasing lambda when necessary. It also influences the rate at which the algorithm converges.

Min Iterations

This sets the minimum number of iterations that the nonlinear regression algorithm must run before it can be terminated. Sometimes, poor starting values make the algorithm think it is finished when it is not. By setting this value to a reasonable number such as 6 or 8, the algorithm is forced to continue even when it thinks it should finish. This reduces the chance of early termination.

Differential Evolution Options

This set of options controls the differential evolution search algorithm.

Max Generations

Specify the maximum number of iterations used by the differential evolution algorithm. Usually, a value between 100 and 200 is adequate.

Hybrid Appraisal Models (Old Version)

Individuals

This is the population size (number of trial points) used by the differential evolution algorithm at each iteration. A value between 15 and 25 is recommended. More points may dramatically increase the running time. Fewer points may not allow the algorithm to converge.

Inheritance

This value controls the amount of movement of each individual toward the current best. Usually, a value between 0.5 and 1.0 is used. We suggest 0.85. A larger value accelerates movement toward the current best, but reduces the chance of locating the global maximum. A smaller value improves the chances of finding the global, rather than a local, solution, but increases the number of iterations until convergence.

Mutation Rate

This value controls the mutation rate of the differential evolution algorithm. This is the probability that a random adjustment is made of a coefficient—which is a *mutation* in the algorithm. Values between 0 and 1 are allowed. A value of 0.3 is recommended.

Grid Range

This is the initial range about each of the initial coefficients that is sampled during the differential evolution algorithm. The algorithm is not limited to this range, but specifying a value large enough to include the solution will increase the probability of convergence.

Min Percent

This option stops the differential evolution iterations when the objective function, defined in terms of absolute percent error, is lower than this amount.

Min Amount

This option stops the differential evolution iterations when the objective function, defined in terms of absolute error, is lower than this amount.

Seed

This option specifies a random seed for the random number generator used by the differential evolution algorithm. Possible values are all integers between 1 and 32000. If you want to obtain the same results, use the same seed value. If you want to let the program select a random seed based on the time-of-day, enter 'RANDOM SEED'.

'Min Start Max' Default Options

Default 'Min Start Max' for Type = (Amounts, Discrete, and Rate)

Enter the default values to be used for the 'Min Start Max' settings when the word 'Defaults' is entered for that option. A separate set of defaults is required for each variable type. Suggested values are:

- (A)mount: '0.001 1 B'
- (D)iscrete: '0 1 5'
- (R)ate: '-5 0 5'

Reports Tab

The following options control which reports are displayed.

Specify Reports

Run Summary Report - Exception Report

Each of these options specifies whether the indicated report is displayed.

Report Options

Exception Percentage

Only rows whose percent prediction error is greater than this will be displayed in the Exception Report.

Precision

Specify the precision of numbers in the report. Single precision will display seven-place accuracy, while the double precision will display thirteen-place accuracy. Note that all reports are formatted for single precision.

Ratio and Sales Decimals

These options control how many decimal places are displayed on the Ratio and Sales values that are shown on the reports.

Factor Labels

These options control the labels that are displayed for each factor. You might want to change them for specific studies. Note that the names are completely arbitrary—they are only used to make the output more readable.

Storage Tab

The predicted values, residuals, and prediction ratios may be stored on the current dataset for further analysis. This group of options lets you designate which statistics (if any) should be stored and which columns should receive these statistics. The selected statistics are automatically stored to the current dataset while the program is executing.

Note that the columns you specify must already have been named on the current dataset.

Note that existing data is replaced. Be careful that you do not specify columns that contain important data.

Storage Columns

Store Predicted Values in Column

The predicted values (\hat{Y}) are stored in this column, if a column is selected.

Store Residuals in Column

The residuals ($Y - \hat{Y}$) are stored in this column, if a column is selected.

Store Ratios in Column

The sales ratios Y/\hat{Y} are stored in this column.

Example 1 – Hybrid Appraisal Model

This section presents an example of how to estimate a prediction equation from the sales price data stored in the Assess dataset. In this example, a hybrid model of the form

$$\text{Overall(Land+Building+Garage)}$$

will be estimated.

You may follow along here by making the appropriate entries or load the completed template **Example 1** by clicking on Open Example Template from the File menu of the Hybrid Appraisal Models window.

1 Open the Assess dataset.

- From the File menu of the NCSS Data window, select **Open Example Dta**.
- Click on the file **Assess.NCSS**.
- Click **Open**.

2 Open the Hybrid Appraisal Models window.

- On the menus, select **Analysis**, then **Mass Appraisal**, then **Hybrid Appraisal Models**. The Hybrid Appraisal Models procedure will be displayed.
- On the menus, select **File**, then **New Template**. This will fill the procedure with the default template.

3 Specify the model.

- On the Hybrid Appraisal Models window, select the **Variables 1 tab**.
- Set the **Estimation Method** to **Minimize the AVERAGE [PERCENT error]**.
- Set the **Y - Dependent Variable (Sales Price)** box to **Sale**.

4 Specify the overall factor.

- On the first line, set **Factor** to **Ov**, **Variable** to **Date**, and **Type** to **R**.
- On the next line, set **Factor** to **Ov**, **Variable** to **Neighborhood(5)**, and **Type** to **D**. The “5” in parentheses will cause indicator variables to be generated for all neighborhoods except neighborhood 5.

5 Specify the building factor.

- On the next line, set **Factor** to **B1**, **Variable** to **GradeLinear**, and **Type** to **R**.
- On the next line, set **Factor** to **B1**, **Variable** to **SqFt1stFlr-Baths**, and **Type** to **A**.

6 Specify the land factor.

- On the next line, set **Factor** to **Ln**, **Variable** to **LotAdjusted**, and **Type** to **R**.
- On the next line, set **Factor** to **Ln**, **Variable** to **LotSize**, and **Type** to **A**.

7 Specify the garage factor.

- On the next line, set **Factor** to **B2**, **Variable** to **GarageSqFt**, and **Type** to **A**.

8 Specify the options.

- Select the **Options tab**.
- Set the **Max Generations** to **50**.
- Set the **Min Percent** to **10**.
- Set the **Seed** to **12346**. Note that normally, you would set this to **RANDOM SEED**.

9 Specify the reports.

- Select the **Reports tab**.
- Check all reports.
- Set the **Exception Percentage** to **25**.

Hybrid Appraisal Models (Old Version)

10 Specify the labels.

- Select the **Reports** tab.
- Set the **Ov** label to **Overall**.
- Set the **Ln** label to **Land**.
- Set the **B1** label to **Building**.
- Set the **B2** label to **Garage**.

11 Run the procedure.

- From the Run menu, select **Run Procedure**. Alternatively, just click the green Run button.

Run Summary Report

Item	Value
Model	Sale=Overall(Land+Building+Garage)
Estimation Method	Average Percent Error (AAPE)
Final Value of AAPE	11.77
R-Squared (from NonLinReg)	0.8394111
Random Number Seed	-12346
Number of Variables Used	10
Number of Parameters in Model	10
Number of Rows Used	76
Number of N.R. Iterations	10
Number of D.E. Iterations	50

This report displays summary information about the analysis such as the model that was fit, the number of rows and variables, the number of iterations, and the random number seed.

Model

This shows the model that was estimated. It gives you a quick overview.

Estimation Method

This is the estimation method that was used.

Final Value of AAPE

This shows the final (minimum) value of the objective function.

R-Squared (from NonLinReg)

This is the R-Squared that was achieved by the nonlinear regression routine. There is no direct R-Squared defined for nonlinear regression. This is a pseudo R-Squared constructed to approximate the usual R-Squared value used in multiple regression. We use the following generalization of the usual R-Squared formula:

$$R\text{-Squared} = (ModelSS - MeanSS)/(TotalSS - MeanSS)$$

where

MeanSS is the sum of squares due to the mean, *ModelSS* is the sum of squares due to the model, and *TotalSS* is the total (uncorrected) sum of squares of Y (the dependent variable).

This version of R-Squared tells you how well the model performs after removing the influence of the mean of Y. Since many nonlinear models do not explicitly include a parameter for the mean of Y, this R-Squared may be negative (in which case we set it to zero) or difficult to interpret. However, if you think of it as a direct extension of the R-Squared that you use in multiple regression, it will serve well for comparative purposes.

Random Number Seed

The random number seed is shown so that if you want to duplicate these results, you can be entering this random number seed.

Hybrid Appraisal Models (Old Version)

Number of Variables Used

This is the number of variables from the database that were used in the analysis.

Number of Parameters in Model

This is the number of parameters that were estimated by the model.

Number of Rows Used

This is the number of rows from the database that were used in the analysis.

Number of N.R. Iterations

This is the number of iterations used by the nonlinear regression procedure.

Number of D.E. Iterations

This is the number of iterations used by the differential evolution algorithm.

Nonlinear Regression Iteration Section

Itn No.	Sum of Squared Errors	Message
0	87346.32	
1	20104.99	
2	5566.038	
3	5076.756	
4	5044.927	
5	5044.661	
6	5044.651	
7	5044.647	
8	5044.647	
Convergence criterion met.		

This report displays the sum of squared errors which is the objective function of the nonlinear regression routine. It allows you to observe the algorithm's progress toward the solution. If you do not see the message 'Convergence criterion met' at the bottom, it means that the algorithm did not terminate normally and you should take corrective action—which usually means that you should increase the maximum number of iterations or provide different starting values.

Differential Evolution Iteration Section

Itn No.	Objective Function (AAPE)
0	12.08115
1	12.08115
2	12.08115
3	12.08115
4	12.08115
5	12.08115
10	12.08115
15	12.08115
20	11.93821
25	11.92409
30	11.80285
35	11.78862
40	11.77882
45	11.77437
50	11.76717

This report displays the value of the objective function that is being minimum by the differential evolution algorithm. In this example, it is the average absolute percent error between the actual and predicted sales price.

Hybrid Appraisal Models (Old Version)

This allows you to see the progress of the algorithm towards a solution. If it looks like the objective function is still shrinking, you may want to rerun the procedure with a larger value for the Max Generations parameter.

Model Specification and Estimation Section

Parm Name	Variable Type	Variable Name	Parameter Estimate	Starting Value	Parameter Bounds
B1	Overall Rate	Date	4.662152E-02	0	-5 to 5
B2	Overall Binary	Neighborhood=4	0.9451685	1	0 to 5
B3	Overall Binary	Neighborhood=6	1.006669	1	0 to 5
B4	Building Rate	GradeLinear	1.776111	0	-5 to 5
B5	Building Amount	SqFt1stFlr	2.225528	1	0.001 to B
B6	Building Amount	SqFtOthFlr	1.634559	1	0.001 to B
B7	Building Amount	Baths	5.391271	1	0.001 to B
B8	Land Rate	LotAdjusted	3.064366	0	-5 to 5
B9	Land Amount	LotSize	0.4990667	1	0.001 to B
B10	Garage Amount	GarageSqFt	2.325238	1	0.001 to B

This report displays the details of the model that was fit.

Parm Name

The name of the parameter shown on this line.

Variable Type

This shows the type of the variable as well as the factor that it belongs to.

Variable Name

The name of the variable on the database. Note that for discrete variables that were expanded into a set of binary indicator variables, a separate line is given for each generated variable. The value indicated by this binary variable is shown after an equals sign. For example, a binary variable was generated that is one when the Neighborhood value is '4' and zero otherwise. This variable is called 'Neighborhood=4'.

Parameter Estimate

This is the estimated value of the parameter in the hybrid model. Note that these values should not be analyzed separately, but together as a group. If you change the model in any way (such as including other variables), these values will change—perhaps substantially!

If you are going to use these values to predict sales prices, you should use the double-precision version of these numbers. These are obtained by setting the Precision value to 'Double' in the Report tab.

Starting Value

These are the values used by the nonlinear regression algorithm in the first iteration. Since the differential evolution algorithm uses the results of a nonlinear regression as its starting values, these values have little influence on the results of the differential evolution algorithm.

Parameter Bounds

These are the limits that were provided for the parameter estimates. If you notice an estimate that is equal to one of its bounds, you should analyze the situation carefully to determine if bound should be relaxed to allow the parameter a wider range.

Hybrid Appraisal Models (Old Version)

Model

Factor	Detail
Overall	Date ^(B1) * (B2) ^(Neighborhood=4) * (B3) ^(Neighborhood=6)
Land	LotAdjusted ^(B8) * ((B9) * LotSize)
Building	GradeLinear ^(B4) * ((B5) * SqFt1stFlr + (B6) * SqFtOthFlr + (B7) * Baths)
Garage	(B10) * GarageSqFt

This report displays the variables that make up each factor. The parameters that will be estimated are B1, B2, ...

Factor

The name of the factor in the hybrid equation.

Detail

The details of the construction of the factor.

Estimated Model

Estimated Model

```
Date^(4.66215216533015E-02) *(0.945168518786127)^(Neighborhood=4)
*(1.00666919375535)^(Neighborhood=6)*((LotAdjusted^(3.0643663225902) *((0.499066748801895)
*LotSize))+(GradeLinear^(1.77611087858318) *((2.22552851585402) *SqFt1stFlr +(1.63455938516463) *SqFtOthFlr
+(5.39127079611148) *Baths))+((2.32523766293606) *GarageSqFt))
```

This is the model with the parameter names replaced with the parameter estimates. This expression may be copied onto the Clipboard and pasted into the transformation section of the database to allow you to predict other observations. Note that this expression is always provided in double precision.

Assessment Ratio Section

Statistic Name	Actual Sale	Predicted Sale	Ratio	Percent Error
Number of Cases	76	76	76	76
Mean	55.9	55.1	1.01	11.77
Minimum	22.5	29.3	0.77	0.00
Lower Quartile	41.7	42.0	0.90	2.87
Median	51.9	49.4	0.99	9.43
Upper Quartile	64.3	62.6	1.05	16.77
Maximum	117.5	130.4	1.58	57.73
Range	95.0	101.1	0.80	57.72
I. Q. Range	22.6	20.6	0.15	13.90
Variance	418.8	379.8	0.03	150.84
Std. Deviation	20.5	19.5	0.17	12.28
Ave Dev. from Median	14.7	13.5	0.12	8.39
Coef. of Variation x 100	36.62	35.37	16.92	104.37
Coef. of Dispersion x 100	28.38	27.31	11.87	88.96
Weighted Mean			0.99	
Price Related Differential			1.02	

This report provides information that assessors have found useful in analyzing the performance of the estimated model. The four columns of the report represent the actual sales price, the predicted sales, and the ratio of the two (predicted over actual), and the percent error in the predicted. Most of the statistics are defined in the Descriptive Statistics procedure. Uncommon terms are defined next.

Ave |Dev - Median|

The average of the absolute values of the deviations of the variable from its median.

Hybrid Appraisal Models (Old Version)

Coef. of Dispersion (COD)

This is 100 times the average absolute deviation about the median divided by the median.

IAOO standards recommend that for single-family residences, COD's of the ratios should be 15.0 or less.

Coef. of Variation (COV)

This is 100 times the standard deviation divided by the mean.

Weighted Mean

The weight ratio mean is the mean of the predicted values divided by the mean of the actual values.

Price Related Differential (PRD)

The price related differential is the mean ratio divided by the weighted mean ratio. It provides an a measure of assessment regressivity or progressivity. A PRD greater than 1.0 indicates that the more expensive properties are underappraised. A PRD less than one indicates that the more expensive properties are overappraised. Experience indicates that this value is normal when it is in the range 0.98 to 1.03.

Predicted Values of New Rows Section

Row No.	Predicted Sale
77	86.3
78	44.6
79	82.4

The section shows the predicted sales price for rows in which values for all variables except the sales price are given.

Using the Model to Predict for New Parcels

You can use your model to predict sales for new values of the model variables. Here is how. Add new rows to the bottom of your database containing the values of the independent variables that you want to create predictions from. Leave the sales price variable blank. When the program analyzes your data, it will skip these rows during the estimation phase, but it will generate predicted values for all rows with a complete set of independent variables, regardless of whether the sales price variable is available.

Exception Report Section

Row No.	Actual Sale	Predicted Sale	Actual-Predicted (Residual)	Predicted/Actual (Ratio)	Percent [Error]
3	22.5	29.6	-7.1	1.32	31.62
23	26.0	38.7	-12.7	1.49	48.89
26	78.5	106.2	-27.7	1.35	35.29
47	33.0	52.1	-19.1	1.58	57.73
53	48.0	67.8	-19.8	1.41	41.35
64	36.0	50.7	-14.7	1.41	40.81
75	30.0	44.6	-14.6	1.49	48.60

This report shows those rows that had a large percentage prediction error. These are the parcels that were poorly predicted by the model. You should analyze them to determine is there is some explanation as to why they were not fit well. You may find that the explanation is as simple as an error in data entry. It may be worth while to rerun the analysis without these rows, especially if there is a reasonable explanation as to why they did not fit the pattern shown by most of the data.

Note that the actual cutoff value for inclusion on this report is set in the Exception Percentage box under the Reports tab.

Hybrid Appraisal Models (Old Version)

Predicted Values and Residuals Section

Row No.	Actual Sale	Predicted Sale	Actual-Predicted (Residual)	Predicted/Actual (Ratio)	Percent [Error]
1	26.0	30.5	-4.5	1.17	17.38
2	53.0	43.8	9.2	0.83	17.31
3	22.5	29.6	-7.1	1.32	31.62
4	85.0	74.3	10.7	0.87	12.57
5	48.0	44.2	3.8	0.92	7.87
.
.
.

This reports shows the actual and predicted sales as well as various measures of there disagreement. Assessors commonly study the Ratio and the Percent Error for individual predictions to determine the goodness of a mass appraisal.