Data modeling: CSCI E-106

Applied Linear Statistical Models

Chapter 13 – Introduction to Nonlinear Regression and Neural Networks

Linear and Nonlinear Regression Models

- the general linear regression model (6.7):

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_{p-1} X_{i,p-1} + \varepsilon_i$$

- A polynomial regression model in one or more predictor variables is linear in the parameters. Ex:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i1}^2 + \beta_3 X_{i2} + \beta_4 X_{i2}^2 + \beta_5 X_{i1} X_{i2} + \varepsilon_i$$

$$\log_{10} Y_i = \beta_0 + \beta_1 \sqrt{X_{i1}} + \beta_2 \exp(X_{i2}) + \varepsilon_i \quad (\textit{transformed})$$

Linear and Nonlinear Regression Models, cont'd

- In general, we can state a linear regression model as:

$$Y_i = f(\boldsymbol{X}_i, \beta) + \varepsilon_i = \boldsymbol{X}_i' \beta + \varepsilon_i$$

where $\boldsymbol{X}_i = [1 \, X_{i1} \, \cdots \, X_{i,p-1}]'$

nonlinear regression models

- the same basic form as that in (13.4):

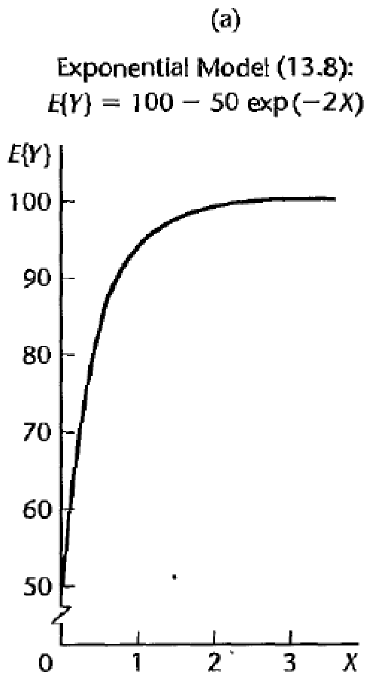
$$Y_i = f(\boldsymbol{X}_i, \gamma) + \varepsilon_i$$

  - $f(\boldsymbol{X}_i, \gamma)$: mean response given by the nonlinear response function $f(\boldsymbol{X}, \gamma)$
  - ε_i: error term; assumed to have $E\{\varepsilon_i\} = 0$, constant variance and to be uncorrelated
  - γ: parameter vector

Linear and Nonlinear Regression Models, cont’d

exponential regression model: in growth studies; concentration

$$Y_i = \gamma_0 \exp(\gamma_1 X_i) + \varepsilon_i, \quad \varepsilon_i \overset{indep.}{\sim} N(0, \sigma^2)$$



(a)

Exponential Model (13.8):
$E\{Y\} = 100 - 50\exp(-2X)$

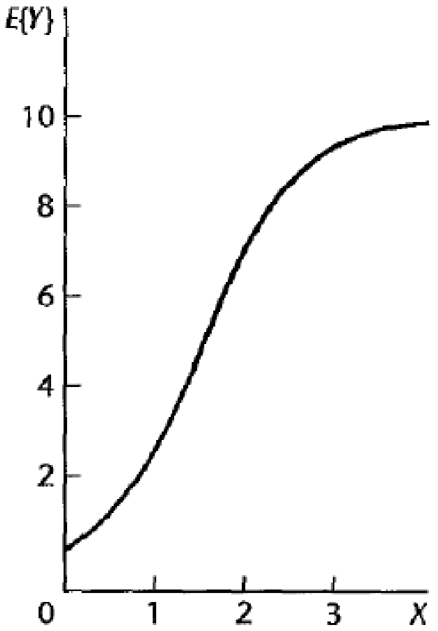
Linear and Nonlinear Regression Models, cont'd

logistic regression models: in population studies

$$Y_i = \frac{\gamma_0}{1 + \gamma_1 \exp(\gamma_2 X_i)} + \varepsilon_i, \quad \varepsilon_i \overset{indep.}{\sim} N(0, \sigma^2)$$

(b)

Logistic Model (13.10):
$E\{Y\} = 10/[1 + 20 \exp(-2X)]$



Linear and Nonlinear Regression Models, cont'd

logistic regression models:

- the response variable is qualitative (0,1): purchase a new car
- the error terms are not normally distributed with constant variance (Chap. 14)

Linear and Nonlinear Regression Models, cont'd

General Form of Nonlinear Regression Models:

- The error terms ε_i are often assumed to be independent normal random variables with constant variance.
- Important difference: $\#\{\beta_i\}$ is not necessarily directly related to $\#\{X_i\}$ in the model
    - linear regression: $(p-1)$ X variables $\Rightarrow$ p regression coefficients

$$Y_i = \sum_{j=0}^{p-1} \beta_j X_{ji} + \varepsilon_i$$

    - nonlinear regression:

$$Y_i = \gamma_0 \exp(\gamma_1 X_i) + \varepsilon_i \quad (p = 2, q = 1)$$

Linear and Nonlinear Regression Models, cont'd

Nonlinear regression model

- q: $\#\{X \text{ variables}\}$
- p: $\#\{\text{regression parameters}\}$
- $\boldsymbol{X}_i$: the observations on the X variables without the initial element 1
- The general form of a nonlinear regression model:

$$Y_i = f(\boldsymbol{X}_i, \gamma) + \varepsilon_i$$

$$\underset{q\times 1}{\boldsymbol{X}_i} = \begin{bmatrix} X_{i1} \\ X_{i2} \\ \vdots \\ X_{iq} \end{bmatrix} \qquad \underset{p\times 1}{\gamma} = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{p-1} \end{bmatrix}$$

Linear and Nonlinear Regression Models, cont'd

intrinsically linear response functions: nonlinear response functions can be linearized by a transformation

Example:

$$f(\boldsymbol{X},\gamma) = \gamma_0[\exp(\gamma_1 X)]$$

$$\Rightarrow g(\boldsymbol{X},\gamma) = \log_e f(\boldsymbol{X},\gamma) = \beta_0 + \beta_1 X$$

$$\beta_0 = \log \gamma_0, \quad \beta_1 = \gamma_1$$

Just because a nonlinear response function is intrinsically linear does not necessarily imply that linear regression is appropriate.

(the error term in the linearized model will no longer be normal with constant variance)

Linear and Nonlinear Regression Models, cont’d

Estimation of regression parameters:

1. least squares method
2. maximum likelihood method

- Also as in linear regression, both of these methods of estimation yield the same parameter estimates when the error terms in (13.12) are independent normal with constant variance.

- It is usually not possible to find analytical expression for LSE and MLE for nonlinear regression models.

- numerical search procedures must be used: require intensive computations

LSE in nonlinear regression

- The concepts of LSE for linear regression also extend directly to nonlinear regression models.
- The least squares criterion:

$$Q = \sum_{i=1}^{n} [Y_i - f(\mathbf{X}_i, \gamma)]^2$$

- Q must be minimized with respect to $\gamma_0, \ldots, \gamma_{p-1}$
- A difference from linear regression is that the solution of the normal equations usually requires an iterative numerical search procedure because analytical solutions generally cannot be found.

Solution to Normal Equations

$$Y_i = f(\boldsymbol{X}_i, \gamma) + \varepsilon_i$$

$$\Rightarrow Q = \sum_{i=1}^{n} [Y_i - f(\boldsymbol{X}_i, \gamma)]^2$$

$$\Rightarrow \boldsymbol{g} = \arg\min_{\gamma} Q \quad (\boldsymbol{g} : \text{ the vector of the LSE } g_k)$$

$$\text{(partial derivative of } Q \text{ with respect to } \gamma_k)$$

$$\Rightarrow \frac{\partial Q}{\partial \gamma_k} = \sum_{i=1}^{n} -2[Y_i - f(\boldsymbol{X}_i, \gamma)] \left[\frac{\partial f(\boldsymbol{X}_i, \gamma)}{\partial \gamma_k} \right] \Bigg|_{\gamma_k = g_k} \overset{set}{=} 0$$

Solution to Normal Equations, cont'd

- The p normal equations:

$$\sum_{i=1}^{n} Y_i \left[\frac{\partial f(\mathbf{X}_i,\gamma)}{\partial \gamma_k}\right]_{\gamma=\boldsymbol{g}} - \sum_{i=1}^{n} f(\mathbf{X}_i,\boldsymbol{g}) \left[\frac{\partial f(\mathbf{X}_i,\gamma)}{\partial \gamma_k}\right]_{\gamma=\boldsymbol{g}} = 0,$$

$$k = 0, 1, \ldots, p-1$$

- $\boldsymbol{g}$: the vector of the least squares estimates g_k

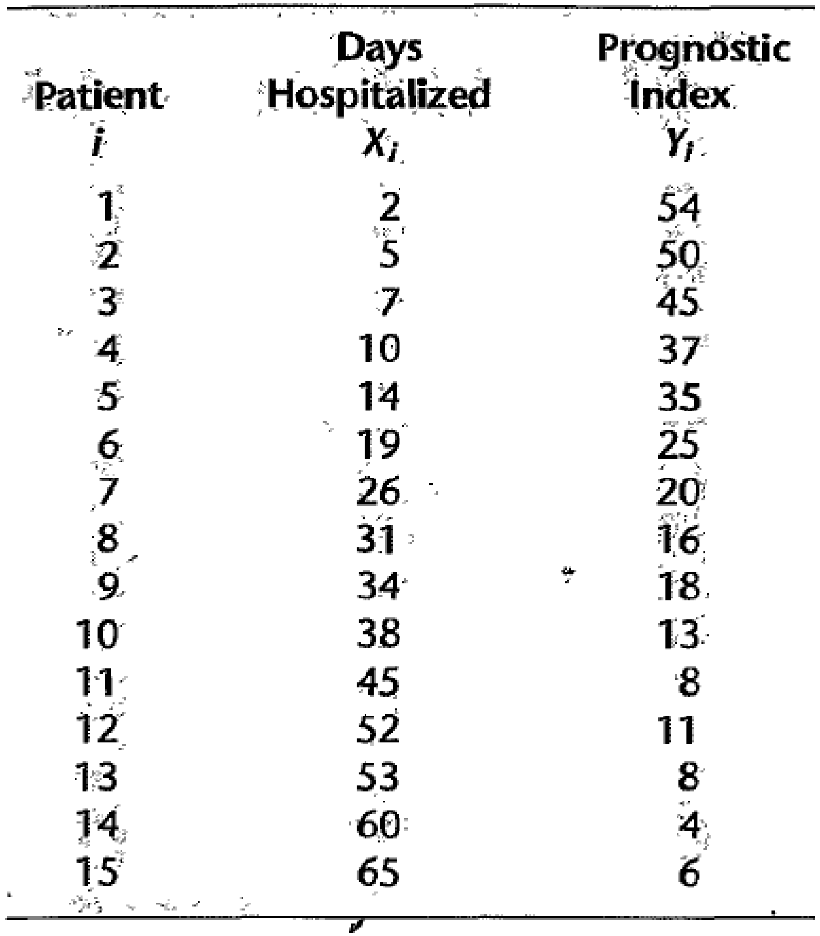
$$\underset{p\times 1}{\boldsymbol{g}} = [g_0, \ \ldots, \ g_{p-1}]'$$

- nonlinear in the parameter estimates g_k
- numerical search procedure are required
- multiple solution may be possible

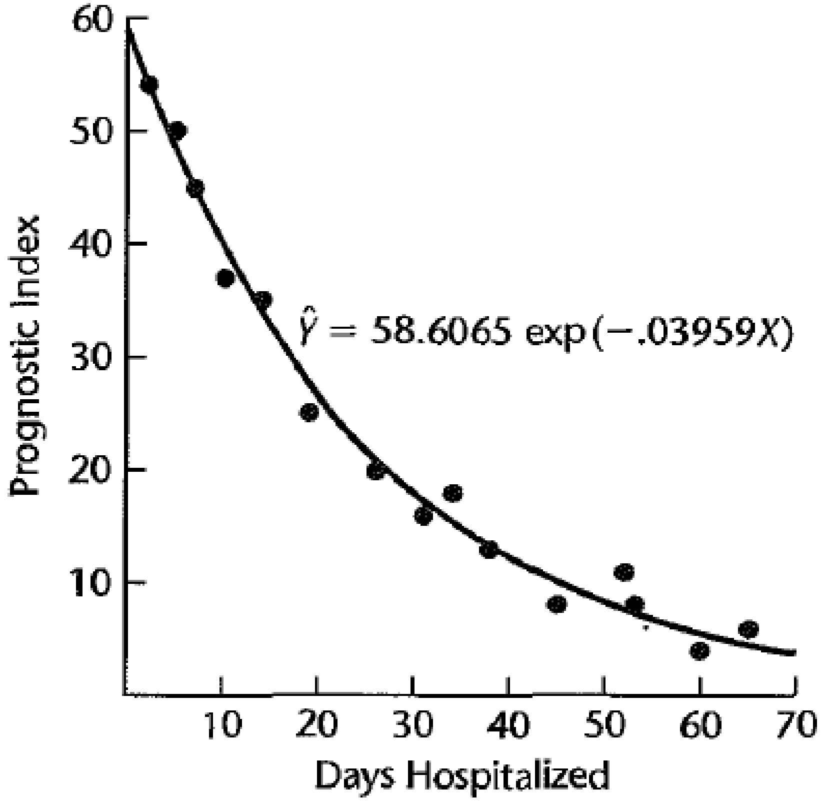
Example

Related earlier studies: the relationship between *Y* and *X* is exponential

$$Y_i = \gamma_0 \exp(\gamma_1 X_i) + \varepsilon_i$$



Patient i	Days Hospitalized X_i	Prognostic Index Y_i
1	2	54
2	5	50
3	7	45
4	10	37
5	14	35
6	19	25
7	26	20
8	31	16
9	34	18
10	38	13
11	45	8
12	52	11
13	53	8
14	60	4
15	65	6



Example, cont'd

$$Q = \sum_{i=1}^{n} [Y_i - \gamma_0 \exp(\gamma_1 X_i)]^2$$

$$\frac{\partial f(\boldsymbol{X}_i, \gamma)}{\partial \gamma_0} = \exp(\gamma_1 X_i)$$

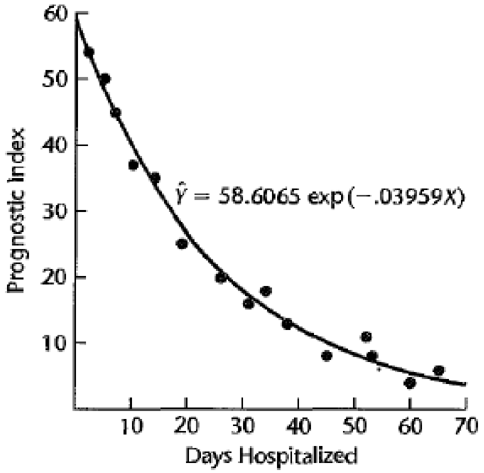
$$\frac{\partial f(\boldsymbol{X}_i, \gamma)}{\partial \gamma_1} = \gamma_0 X_i \exp(\gamma_1 X_i)$$

$$\Rightarrow \frac{\partial Q}{\partial \gamma_k}\Big|_{\boldsymbol{g}} = 0$$

$$\Rightarrow \sum Y_i \exp(g_1 X_i) - g_0 \sum \exp(2g_1 X_i) = 0$$

$$\sum Y_i X_i \exp(g_1 X_i) - g_0 \sum X_i \exp(2g_1 X_i) = 0$$

FIGURE 13.2 Scatter Plot and Fitted Nonlinear Regression Function—Severely Injured Patients Example.



No closed-form solution exists for $\boldsymbol{g} = (g_0, g_1)^T$

Gauss-Newton Method

- linearization method:
  1. use a Taylor series expansion to approximate the nonlinear regression model

$$\left(f(x) = f(a) + \sum_{n=1}^{\infty} \frac{f^{(n)}}{n!}(x-a)^n\right)$$

  2. employ OLS to estimate the parameters

Gauss-Newton Method, cont'd

Gauss-Newton Method:

1. initial parameters $\gamma_0, \ldots, \gamma_{p-1}$: $g_0^{(0)}, \ldots, g_{p-1}^{(0)}$
2. approximate the mean responses $f(\boldsymbol{X}_i, \gamma)$ in the Taylor series expansion around $g_k^{(0)}$:

$$f(\boldsymbol{X}_i, \gamma) \approx f(\boldsymbol{X}_i, \boldsymbol{g}^{(0)}) + \sum_{k=0}^{p-1} \left[\frac{\partial f(\boldsymbol{X}_i, \gamma)}{\partial \gamma_k} \right]_{\gamma = \boldsymbol{g}^{(0)}} (\gamma_k - g_k^{(0)})$$

3. obtain revised estimated regression coefficients $g_k^{(1)}$: (later)

$$g_k^{(1)} = g_k^{(0)} + b_k^{(0)}$$

Gauss-Newton Method, cont'd

$$f(\mathbf{X}_i, \gamma) \approx f(\mathbf{X}_i, \mathbf{g}^{(0)}) + \sum_{k=0}^{p-1} \left[\frac{\partial f(\mathbf{X}_i, \gamma)}{\partial \gamma_k} \right]_{\gamma = \mathbf{g}^{(0)}} (\gamma_k - g_k^{(0)})$$

$$= f_i^{(0)} + \sum_{k=0}^{p-1} D_{ik}^{(0)} \beta_k^{(0)}$$

$$\Rightarrow Y_i \approx f_i^{(0)} + \sum_{k=0}^{p-1} D_{ik}^{(0)} \beta_k^{(0)} + \varepsilon_i$$

$$\overset{(Y_i^{(0)} = Y_i - f_i^{(0)})}{\Longrightarrow} Y_i^{(0)} \approx \sum_{k=0}^{p-1} D_{ik}^{(0)} \beta_k^{(0)} + \varepsilon_i \quad \text{(no intercept)} \qquad (13.24)$$

- The purpose of fitting the linear regression model approximation (13.24) is therefore to estimate $\beta_k^{(0)}$ and use these estimates to adjust the initial starting estimates of the regression parameters.

Gauss-Newton Method, cont’d

Matrix Form: $Y_i^{(0)} \approx \sum_{k=0}^{p-1} D_{ik}^{(0)} \beta_k^{(0)} + \varepsilon_i$

$$\mathbf{Y}^{(0)} \approx \mathbf{D}^{(0)} \boldsymbol{\beta}^{(0)} + \boldsymbol{\varepsilon} \tag{13.25}$$

where:

(13.25a) $\underset{n \times 1}{\mathbf{Y}^{(0)}} = \begin{bmatrix} Y_1 - f_1^{(0)} \\ \vdots \\ Y_n - f_n^{(0)} \end{bmatrix}$ **(13.25b)** $\underset{n \times p}{\mathbf{D}^{(0)}} = \begin{bmatrix} D_{10}^{(0)} & \cdots & D_{1,p-1}^{(0)} \\ \vdots & & \vdots \\ D_{n0}^{(0)} & \cdots & D_{n,p-1}^{(0)} \end{bmatrix}$

(13.25c) $\underset{p \times 1}{\boldsymbol{\beta}^{(0)}} = \begin{bmatrix} \beta_0^{(0)} \\ \vdots \\ \beta_{p-1}^{(0)} \end{bmatrix}$ **(13.25d)** $\underset{n \times 1}{\boldsymbol{\varepsilon}} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}$

Gauss-Newton Method, cont’d

- the $\boldsymbol{D}$ matrix of partial derivative play the role of the $\boldsymbol{X}$ matrix (without a column of 1s for the intercept)
- Estimate $\beta^{(0)}$ by OLS:

$$\boldsymbol{b}^{(0)} = (\boldsymbol{D}^{(0)'}\boldsymbol{D}^{(0)})^{-1}\boldsymbol{D}^{(0)'}\boldsymbol{Y}^{(0)}$$

- obtain revised estimated regression coefficients $g_k^{(1)}$:

$$g_k^{(1)} = g_k^{(0)} + b_k^{(0)}$$

Gauss-Newton Method, cont'd

- Evaluated for $\boldsymbol{g}^{(0)}$ by $SSE^{(0)}$:

$$SSE^{(0)} = \sum_{i=1}^{n} [Y_i - f(\boldsymbol{X}_i, \boldsymbol{g}^{(0)})]^2 = \sum_{i=1}^{n} [Y_i - f_i^{(0)}]^2$$

- After the end of the first iteration:

$$SSE^{(1)} = \sum_{i=1}^{n} [Y_i - f(\boldsymbol{X}_i, \boldsymbol{g}^{(1)})]^2 = \sum_{i=1}^{n} [Y_i - f_i^{(1)}]^2$$

- If the Gauss-Newton method is working effectively in the first iteration, $SSE^{(1)}$ should be smaller than $SSE^{(0)}$. ($\because$ $\boldsymbol{g}^{(1)}$ should be better estimates)

Gauss-Newton Method, cont'd

- The Gauss-Newton method repeats the procedure with $\mathbf{g}^{(1)}$ now used for the new starting values.
- Until $\mathbf{g}^{(s+1)} - \mathbf{g}^{(s)}$ and/or $SSE^{(s+1)} - SSE^{(s)}$ become negligible
- The Gauss-Newton method works effectively in many nonlinear regression applications. (Sometimes may require numerous iterations before converging.)

Gauss-Newton Method, cont'd

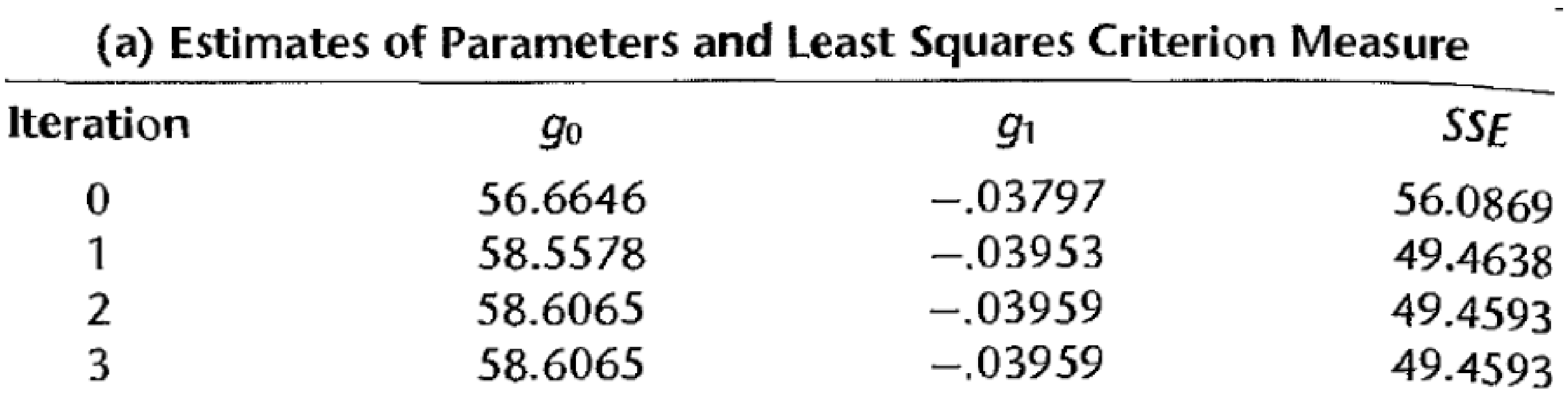
Example: Severely injured patients

- Initial: Transformed Y $\quad (\log \gamma_0 \exp(\gamma_1 X) = \log \gamma_0 + \gamma_1 X)$

$$Y_i' = \beta_0 + \beta_1 X_i + \varepsilon_i$$

$$\overset{OLS}{\Rightarrow} b_0 = 0.40371, \quad b_1 = -0.03797$$

$$\Rightarrow g_0^{(0)} = exp(b_0) = 56.6646, \quad g_1^{(0)} = b_1 = -0.03797$$



(a) Estimates of Parameters and Least Squares Criterion Measure

Iteration	g_0	g_1	SSE
0	56.6646	−.03797	56.0869
1	58.5578	−.03953	49.4638
2	58.6065	−.03959	49.4593
3	58.6065	−.03959	49.4593

Gauss-Newton Method, cont'd

$$f(\mathbf{X}_1, \mathbf{g}^{(0)}) = f_1^{(0)} = g_0^{(0)} \exp(g_1^{(0)} X_1) = (56.6646) \exp[-.03797(2)] = 52.5208$$

Since $Y_1 = 54$, the deviation from the mean response is:

$$Y_1^{(0)} = Y_1 - f_1^{(0)} = 54 - 52.5208 = 1.4792$$

$$\begin{aligned} SSE^{(0)} &= \sum (Y_i - f_i^{(0)})^2 = \sum (Y_i^{(0)})^2 \\ &= (1.4792)^2 + \cdots + (1.1977)^2 = 56.0869 \end{aligned}$$

$$D_{10}^{(0)} = \left[\frac{\partial f(\mathbf{X}_1, \boldsymbol{\gamma})}{\partial \gamma_0}\right]_{\boldsymbol{\gamma}=\mathbf{g}^{(0)}} = \exp(g_1^{(0)} X_1) = \exp[-.03797(2)] = .92687$$

$$\begin{aligned} D_{11}^{(0)} = \left[\frac{\partial f(\mathbf{X}_1, \boldsymbol{\gamma})}{\partial \gamma_1}\right]_{\boldsymbol{\gamma}=\mathbf{g}^{(0)}} &= g_0^{(0)} X_1 \exp(g_1^{(0)} X_1) \\ &= 56.6646(2) \exp[-.03797(2)] = 105.0416 \end{aligned}$$

$$\mathbf{b}^{(0)} = \begin{bmatrix} 1.8932 \\ -.001563 \end{bmatrix}$$

$$\mathbf{g}^{(1)} = \mathbf{g}^{(0)} + \mathbf{b}^{(0)} = \begin{bmatrix} 56.6646 \\ -.03797 \end{bmatrix} + \begin{bmatrix} 1.8932 \\ -.001563 \end{bmatrix} = \begin{bmatrix} 58.5578 \\ -.03953 \end{bmatrix}$$

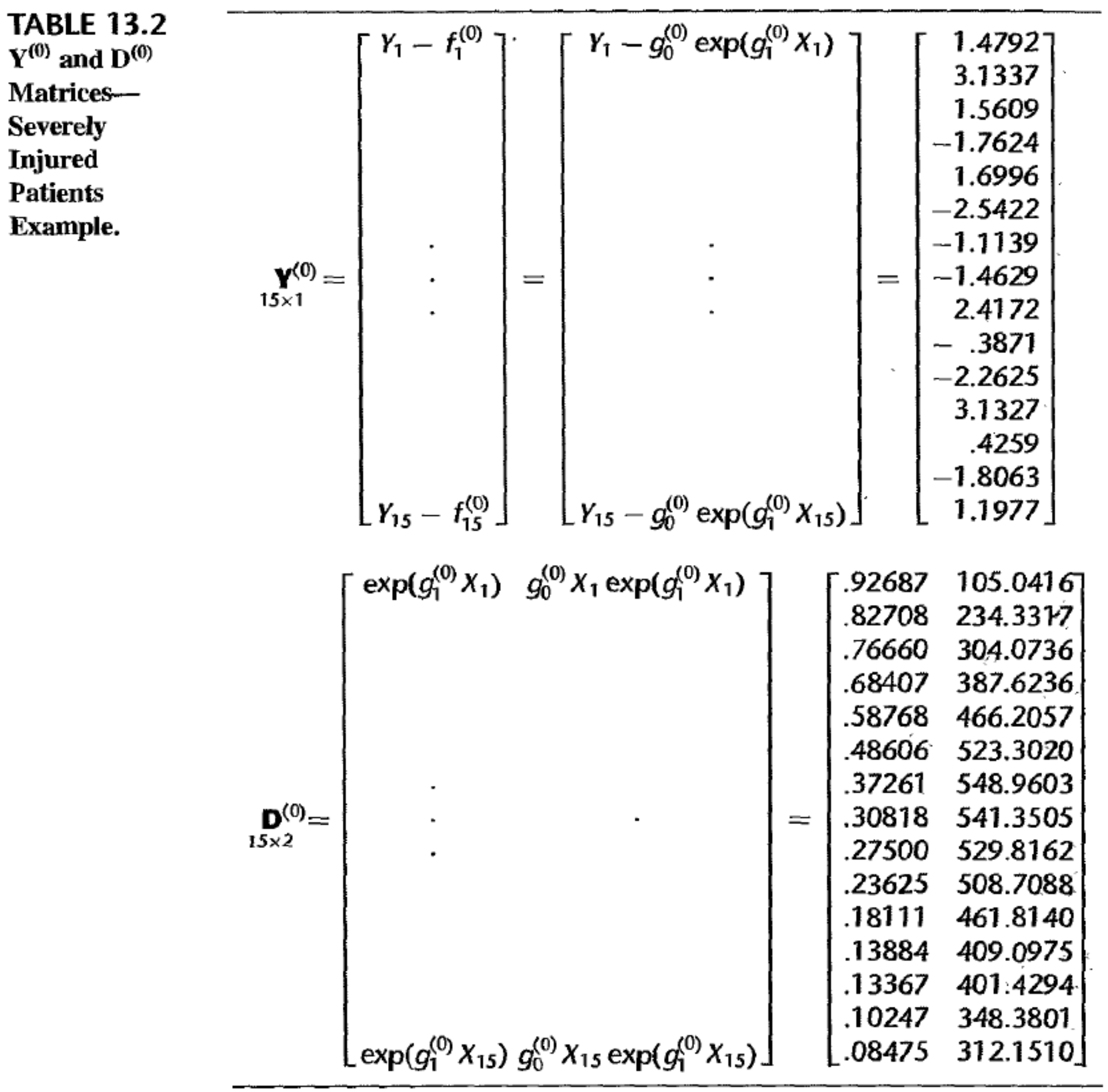
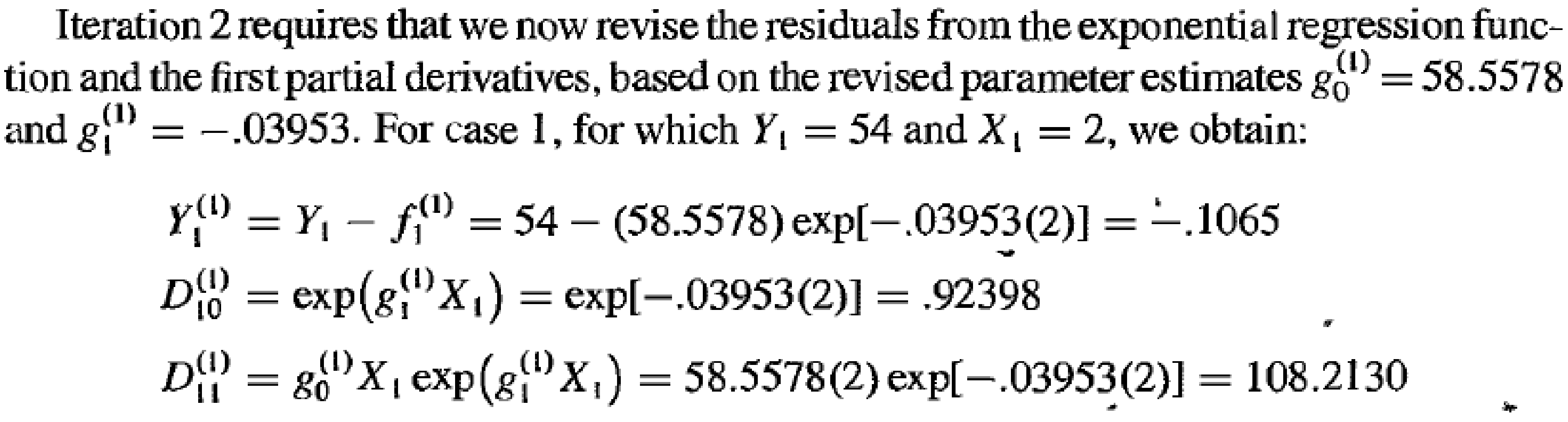


TABLE 13.2 $\mathbf{Y}^{(0)}$ and $\mathbf{D}^{(0)}$ Matrices—Severely Injured Patients Example.

$$\underset{15\times 1}{\mathbf{Y}^{(0)}} = \begin{bmatrix} Y_1 - f_1^{(0)} \\ \cdot \\ \cdot \\ \cdot \\ Y_{15} - f_{15}^{(0)} \end{bmatrix} = \begin{bmatrix} Y_1 - g_0^{(0)} \exp(g_1^{(0)} X_1) \\ \cdot \\ \cdot \\ \cdot \\ Y_{15} - g_0^{(0)} \exp(g_1^{(0)} X_{15}) \end{bmatrix} = \begin{bmatrix} 1.4792 \\ 3.1337 \\ 1.5609 \\ -1.7624 \\ 1.6996 \\ -2.5422 \\ -1.1139 \\ -1.4629 \\ 2.4172 \\ -.3871 \\ -2.2625 \\ 3.1327 \\ .4259 \\ -1.8063 \\ 1.1977 \end{bmatrix}$$

$$\underset{15\times 2}{\mathbf{D}^{(0)}} = \begin{bmatrix} \exp(g_1^{(0)} X_1) & g_0^{(0)} X_1 \exp(g_1^{(0)} X_1) \\ \cdot & \\ \cdot & \cdot \\ \cdot & \\ \exp(g_1^{(0)} X_{15}) & g_0^{(0)} X_{15} \exp(g_1^{(0)} X_{15}) \end{bmatrix} = \begin{bmatrix} .92687 & 105.0416 \\ .82708 & 234.3317 \\ .76660 & 304.0736 \\ .68407 & 387.6236 \\ .58768 & 466.2057 \\ .48606 & 523.3020 \\ .37261 & 548.9603 \\ .30818 & 541.3505 \\ .27500 & 529.8162 \\ .23625 & 508.7088 \\ .18111 & 461.8140 \\ .13884 & 409.0975 \\ .13367 & 401.4294 \\ .10247 & 348.3801 \\ .08475 & 312.1510 \end{bmatrix}$$

Gauss-Newton Method, cont'd

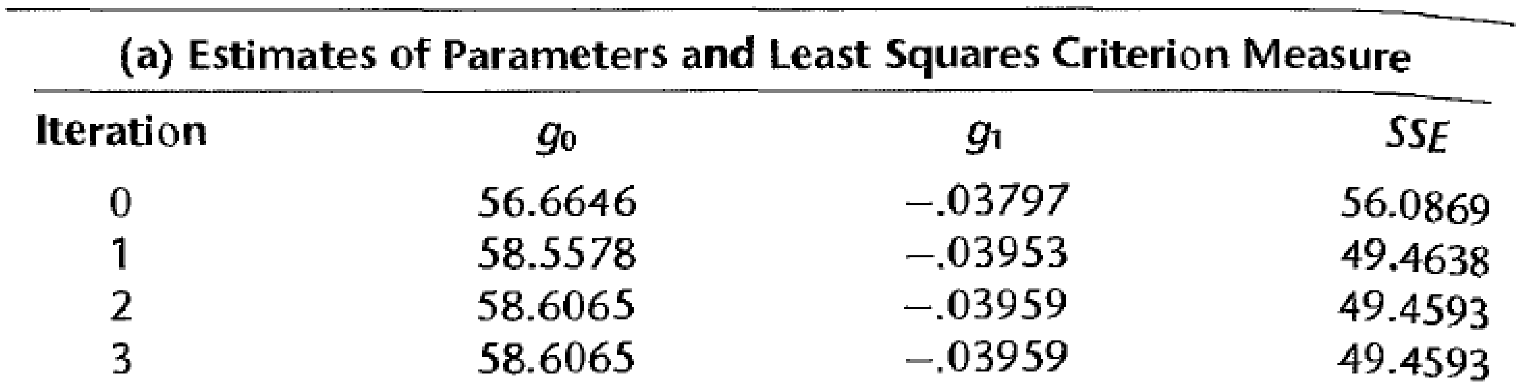


Iteration 2 requires that we now revise the residuals from the exponential regression function and the first partial derivatives, based on the revised parameter estimates $g_0^{(1)} = 58.5578$ and $g_1^{(1)} = -.03953$. For case 1, for which $Y_1 = 54$ and $X_1 = 2$, we obtain:

$$Y_1^{(1)} = Y_1 - f_1^{(1)} = 54 - (58.5578)\exp[-.03953(2)] = -.1065$$

$$D_{10}^{(1)} = \exp\left(g_1^{(1)} X_1\right) = \exp[-.03953(2)] = .92398$$

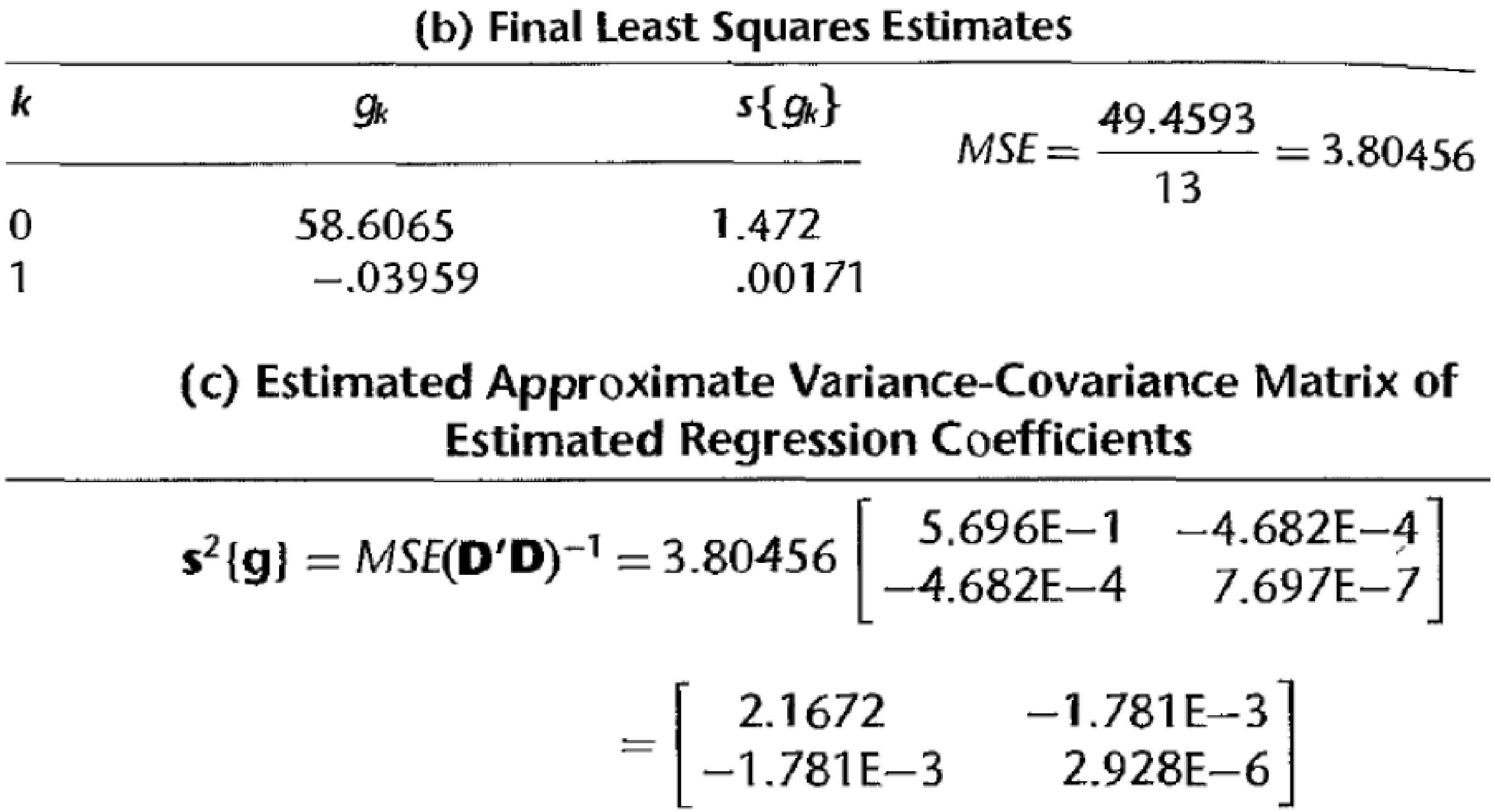
$$D_{11}^{(1)} = g_0^{(1)} X_1 \exp\left(g_1^{(1)} X_1\right) = 58.5578(2)\exp[-.03953(2)] = 108.2130$$



(a) Estimates of Parameters and Least Squares Criterion Measure

Iteration	g_0	g_1	SSE
0	56.6646	−.03797	56.0869
1	58.5578	−.03953	49.4638
2	58.6065	−.03959	49.4593
3	58.6065	−.03959	49.4593

Gauss-Newton Method, cont'd



(b) Final Least Squares Estimates

k	g_k	$s\{g_k\}$
0	58.6065	1.472
1	−.03959	.00171

$$MSE = \frac{49.4593}{13} = 3.80456$$

(c) Estimated Approximate Variance-Covariance Matrix of Estimated Regression Coefficients

$$\mathbf{s}^2\{\mathbf{g}\} = MSE(\mathbf{D}'\mathbf{D})^{-1} = 3.80456 \begin{bmatrix} 5.696\text{E}{-1} & -4.682\text{E}{-4} \\ -4.682\text{E}{-4} & 7.697\text{E}{-7} \end{bmatrix}$$

$$= \begin{bmatrix} 2.1672 & -1.781\text{E}{-3} \\ -1.781\text{E}{-3} & 2.928\text{E}{-6} \end{bmatrix}$$

$$\hat{Y} = (58.6065)\exp(-0.03959X)$$

Gauss-Newton Method, cont'd

- The choice of initial starting values:
  - a poor choice may result in slow convergence, convergence to a local minimum, or even divergence
  - Good starting values: result in faster convergence, will lead to a solution that is the global minimum rather than a local minimum
- A variety of methods for obtaining starting values:
  1. related earlier studies
  2. select p representative observations$\Rightarrow$ solve for p parameters, then used as the starting values
  3. do a grid search in the parameter space$\Rightarrow$ using as the starting values that $\boldsymbol{g}$ for which Q is smallest

Gauss-Newton Method, cont’d

- Some properties that exist for linear regression least squares do not hold for nonlinear regression least squares.
- Ex: $\sum e_i \neq 0$; $SSR + SSE \neq SSTO$; R^2 is not a meaningful descriptive statistic for nonlinear regression.
- Two other direct search procedures:
  1. The method of steepest descent searches
  2. The Marquardt algorithm: seeks to utilize the best feature of the
- Gauss-Newton method and the method of steepest descent a middle ground between these two method

Model building and diagnostic

- The model-building process for nonlinear regression models often differs somewhat from that for linear regression models
- Validation of the selected nonlinear regression model can be performed in the same fashion as for linear regression models.
- Use of diagnostics tools to examine the appropriateness of a fitted model plays an important role in the process of building a nonlinear regression model.

Model building and diagnostic, cont'd

- When replicate observations are available and the sample size is reasonably large, the appropriate of a nonlinear regression function can be tested formally by means of the lack of fit test for linear regression models. (the test is an approximate one)
- Plots: e_i vs. t_i, $\hat{Y}_i$, X_{ik} can be helpful in diagnosing departures from the assumed model
- Unequal variances⇒ WLS; transformations

Inferences

- Inferences about the regression parameters in nonlinear regression are usually based on large-sample theory.
- When n is large, LSE and MLE for nonlinear regression models with normal error terms are approximately normally distributed and almost unbiased and have almost minimum variance
- Estimate of Error Term Variance:

$$MSE = \frac{SSE}{n-p} = \frac{\sum[Y_i - f(\mathbf{X}_i, \mathbf{g})]^2}{n-p}$$

(Not unbiased estimator of σ^2 but the bias is small when n is large)

Inferences, cont'd

Large-Sample Theory

When the error terms ε_i are independent $N(0, \sigma^2)$ and the sample size n is reasonably large, the sampling distribution of $\mathbf{g}$ is approximately normal. The expected value of the mean vector is approximately:

$$E\{\mathbf{g}\} \approx \gamma$$

The approximate variance-covariance matrix of the regression coefficients is estimated by:

$$\mathbf{s}^2\{\mathbf{g}\} = MSE(\boldsymbol{D}'\boldsymbol{D})^{-1}$$

Inferences, cont'd

- No simple rule exists that tells us when it is appropriate to use the large-sample inference methods and when it is not appropriate.
- However, a number of guidelines have been developed that are helpful in assessing the appropriateness of using the large-sample inference procedures in a given application.
- When the diagnostics suggest that large-sample inference procedures are not appropriate in a particular instance, remedial measures should be explored.
    - reparameterize the nonlinear regression model
    - bootstrap estimated of precision and confidence intervals instead of the large-sample inferences

Interval Estimation

- Large-sample theorem: approximate result for a single γ_k

$$\frac{g_k - \gamma_k}{s\{g_k\}} \sim t(n-p), \quad k = 0, 1, \ldots, p-1$$
$$\Rightarrow g_k \pm t(1-\alpha/2; n-p)s\{g_k\}$$

- Several γ_k: m parameters to be estimated with approximate family confidence coefficient $1-\alpha \Rightarrow$ the Bonferroni confidence limits:

$$g_k \pm Bs\{g_k\} \quad B = t(1-\alpha/2m; n-p)$$

Test Concerning single γ_k

- A large-sample test:

$$H_0 : \gamma_k = \gamma_{k0} \text{ vs. } H_a : \gamma_k \neq \gamma_{k0}$$

$$t^* = \frac{g_k - \gamma_{k0}}{s\{g_k\}}$$

- If $|t^*| \leq t(1-\alpha/2; n-p)$, conclude H_0
  If $|t^*| > t(1-\alpha/2; n-p)$, conclude H_a
- Test concerning several γ_k

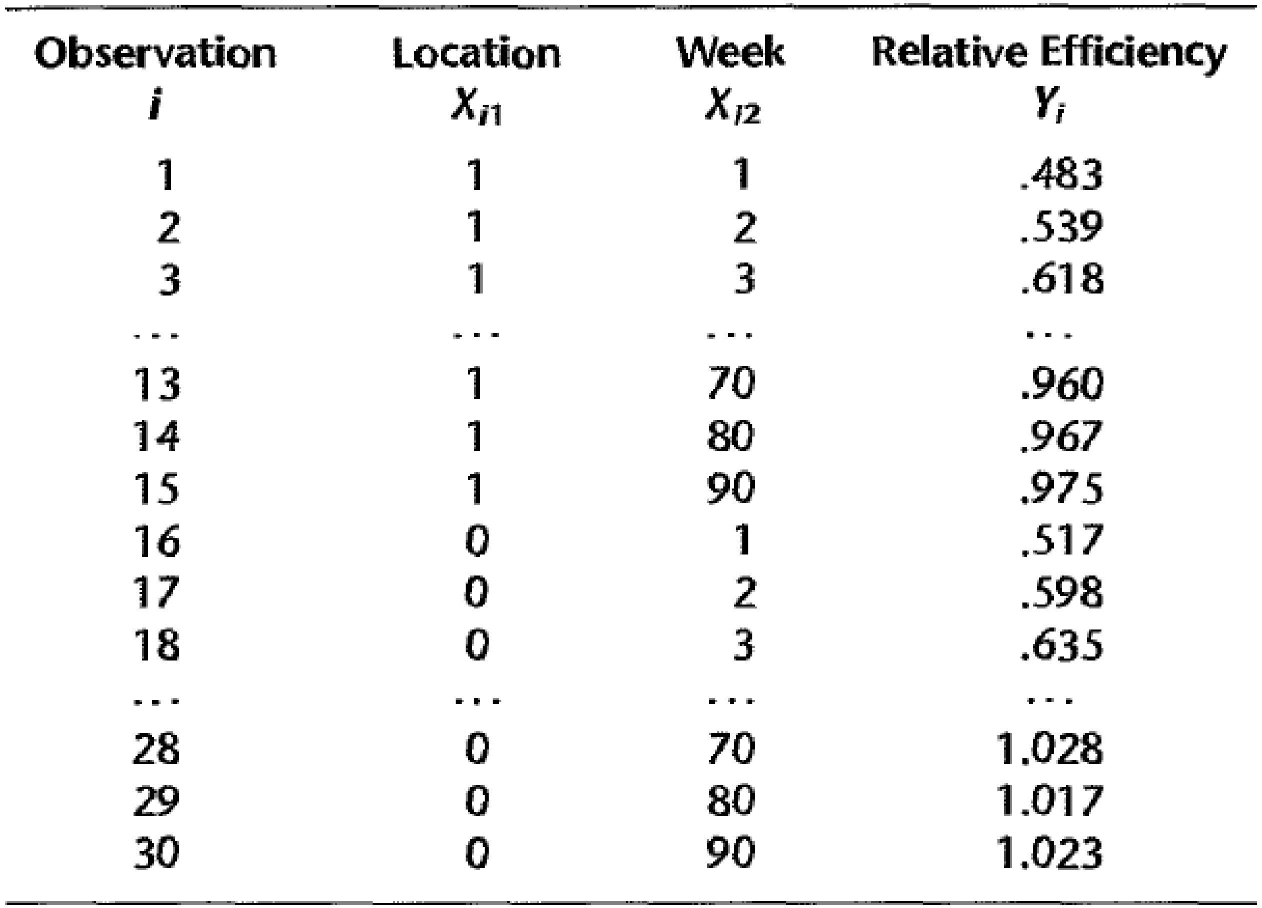
$$F^* = \frac{SSE(R) - SSE(F)}{df_R - df_F} \div MSE(F)$$

$$\overset{approx}{\sim} F(df_R - df_F, df_F) \text{ when } H_0 \text{ holds}$$

Example: Learning Curve

An electronics products manufacturer undertook the production of a new product in two locations (location A: coded X_1 = 1, location B: coded X_1 = 0). Relative efficiency, the response variable (Y) in the study, was calculated for 90 weeks for each location. The model decided on was:

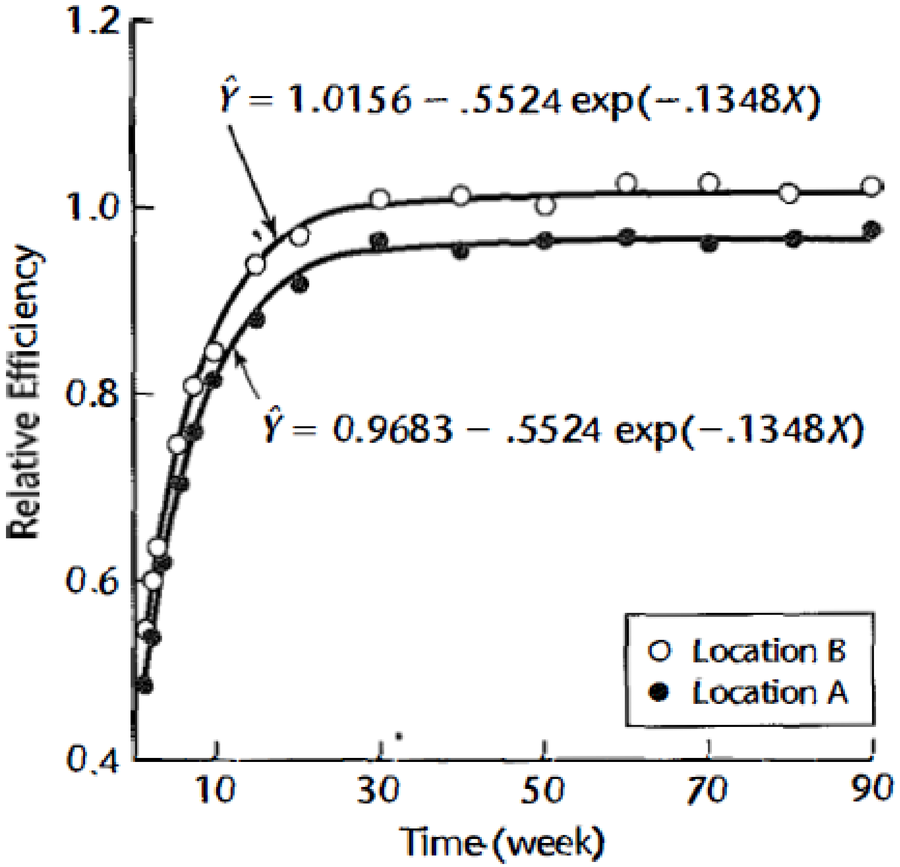
$$Y_i = \gamma_0 + \gamma_1 X_{i1} + \gamma_3 exp(\gamma_2 X_{i2}) + \varepsilon_i$$



Observation i	Location X_{i1}	Week X_{i2}	Relative Efficiency Y_i
1	1	1	.483
2	1	2	.539
3	1	3	.618
...	...	...	...
13	1	70	.960
14	1	80	.967
15	1	90	.975
16	0	1	.517
17	0	2	.598
18	0	3	.635
...	...	...	...
28	0	70	1.028
29	0	80	1.017
30	0	90	1.023

Example: Learning Curve, cont’d

$$Y_i = \gamma_0 + \gamma_1 X_{i1} + \gamma_3 exp(\gamma_2 X_{i2}) + \varepsilon_i$$



Previous studies indicated that:

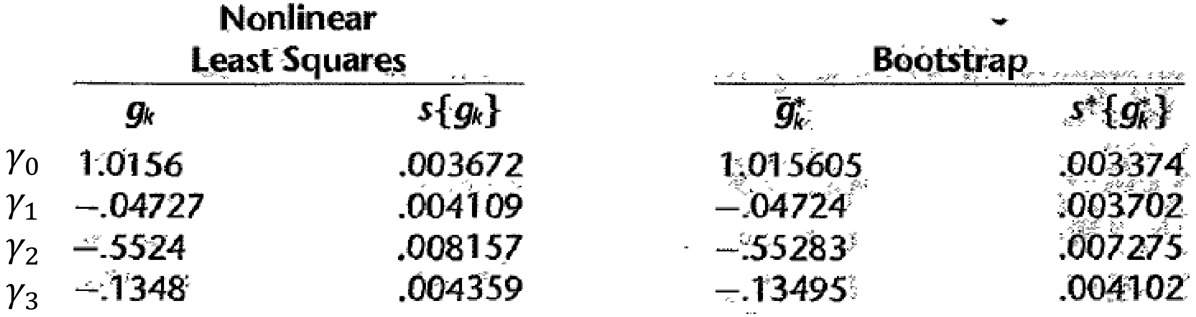
γ_0 is around 1.025 ⇒ $g_0^{(0)} = 1.025$

γ_1 is around -0.0459 ⇒ $g_1^{(0)} = -0.0459$

γ_2 is around -0.0459 ⇒ $g_2^{(0)} = -0.122$

γ_3 is around -0.5 ⇒ $g_3^{(0)} = -0.5$

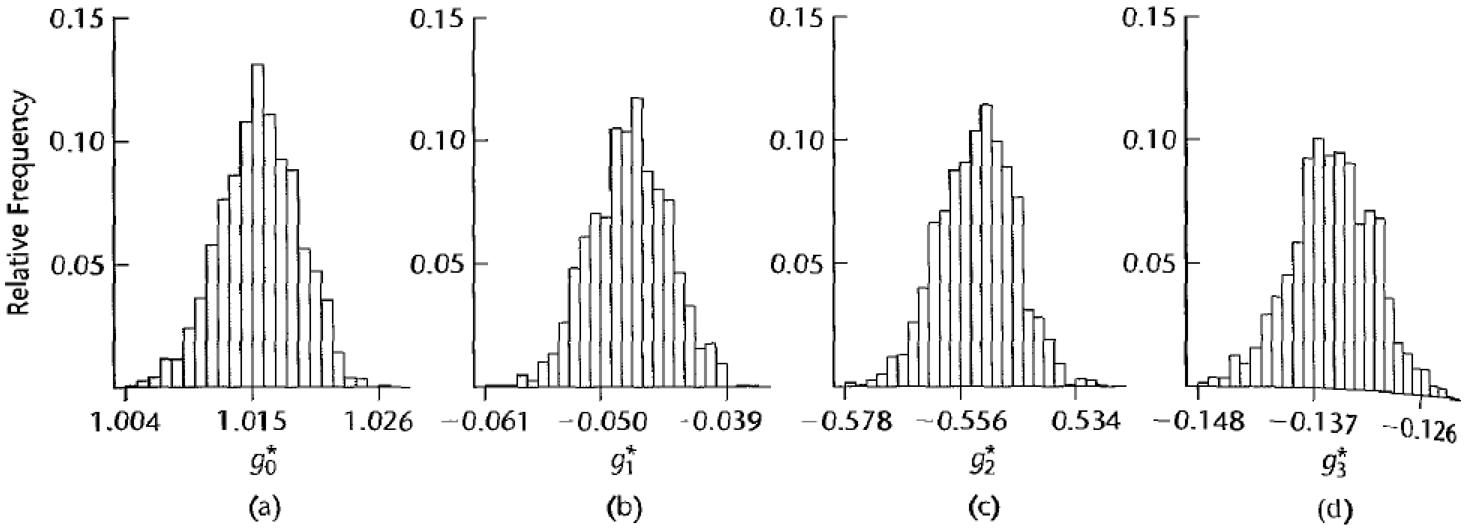
After 5 iteration, the results converges.
Bootstrap results are stated below.



	Nonlinear Least Squares		Bootstrap	
	g_k	$s\{g_k\}$	$\bar{g}_k^*$	$s^*\{g_k^*\}$
γ_0	1.0156	.003672	1.015605	.003374
γ_1	−.04727	.004109	−.04724	.003702
γ_2	−.5524	.008157	−.55283	.007275
γ_3	−.1348	.004359	−.13495	.004102

Example: Learning Curve, cont’d

FIGURE 13.6 MINITAB Histograms of Bootstrap Sampling Distributions—Learning Curve Example.



- Each least squares estimate g_k is very close to the mean $\bar{g}_k$; indicating that the estimates have very little bias.
- Each large-sample standard deviation $s\{g_k\}$ is fairly close to the respective bootstrap standard deviation $s\{g_k^*\}$.
- All 4 histograms appear to be consistent with approximately normal sampling distributions.
- These results all indicate that the sampling behavior of the nonlinear regression estimates is close to linear and therefore support the use of large-sample inferences here.

Example: Learning Curve, cont'd

In the severely injured patients example:

- 95 percent statement confidence interval for γ_1:

t(1-0.05/2,30-4) = t(0.75,26)=2.056

$$-.04727 \pm 2.056(.004109) \quad \Rightarrow -.0557 \leq \gamma_1 \leq -.0388$$

- The joint confidence intervals with approximate family confidence coefficient of 90 percent:

t(1-0.1/(2*2),13) = t(0.975,26)=2.056

$$1.0156 \pm 2.056(.003672) \quad \Rightarrow 1.008 \leq \gamma_0 \leq 1.023$$

$$-.04727 \pm 2.056(.004109) \quad \Rightarrow -0.0557 \leq \gamma_1 \leq -0.0388$$

$$-0.5524 \pm 2.056(.008157) \quad \Rightarrow -0.569 \leq \gamma_2 \leq -0.536$$

$$-0.1348 \pm 2.056(.004359) \quad \Rightarrow -0.144 \leq \gamma_3 \leq -0.126$$

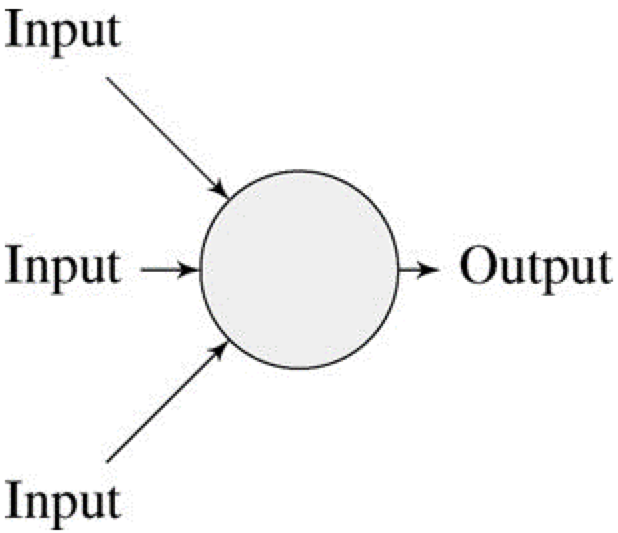
Neural Network Model

The neural network model is simply a nonlinear statistical model that contains many more parameters than the corresponding linear statistical model:

- Typically be overparameterized
- Resulting in parameters that are uninterpretable

Neural Network Model will often perform better in predicting future responses than a standard regression model.

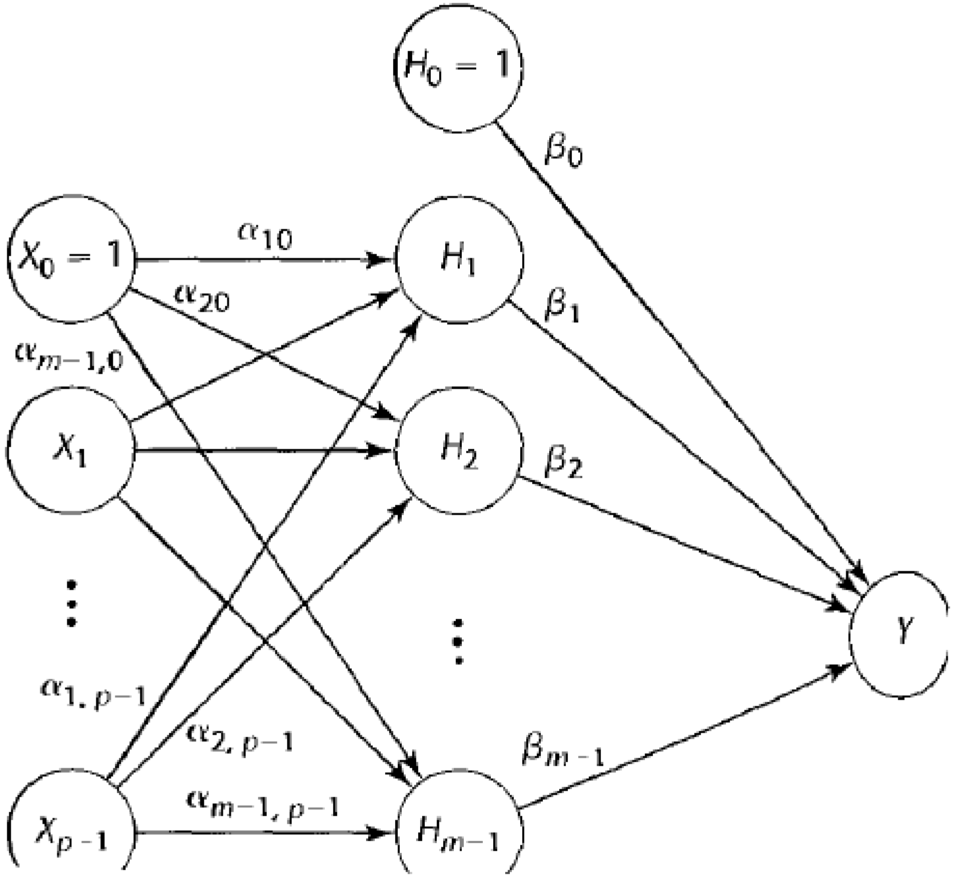
Neural Network Model, cont’d



Neural networks (NN) were originally developed as an attempt to emulate the human brain. The original idea behind neural networks was to use a computer-based model of the human brain to perform complex tasks. We can recognize people in fractions of a second, but this task is difficult for computers. So why not make software more like the human brain?

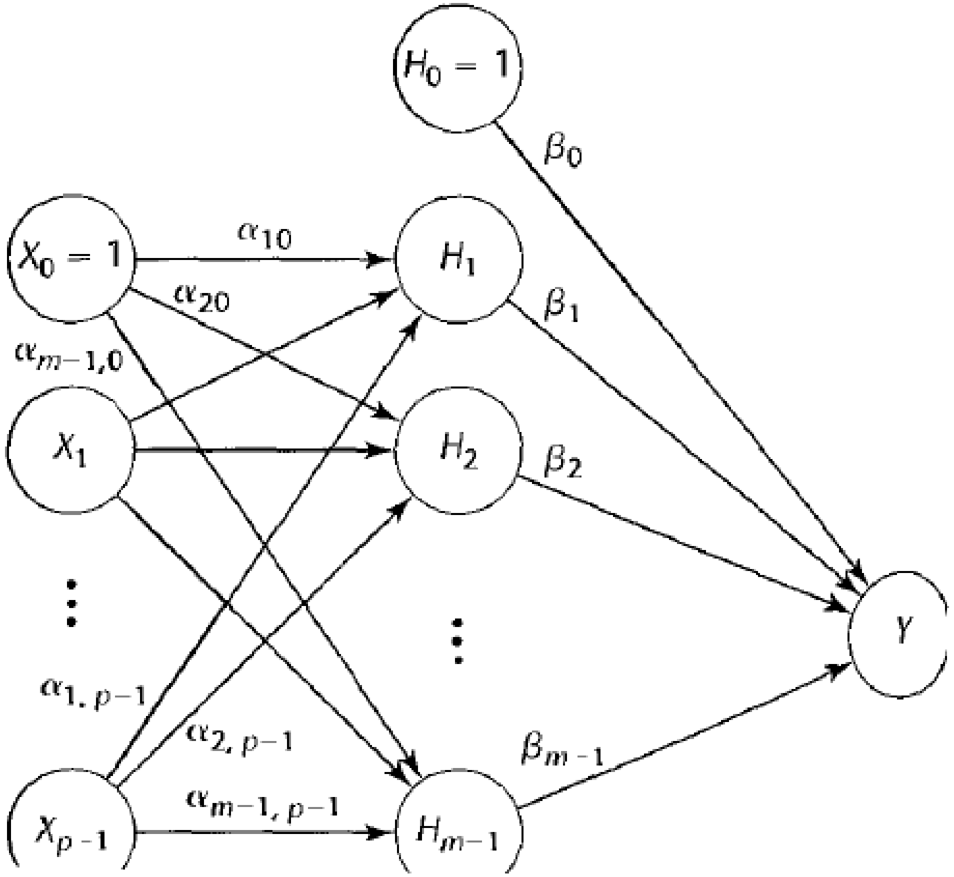
The brain model of connected neurons, first suggested by McCulloch and Pitts (**1943**), is too simplistic given more recent research. For these and other reasons, the methodology is more properly called *artificial* neural nets.

Single-Hidden-Layer Feedforward Neural Network



- The predictor nodes are labeled $X_0, X_1 \cdots, X_{p-1}$
- The hidden nodes are labeled $H_0, H_1 \cdots, H_{m-1}$
- Finally, the hidden nodes are linked to the response *Y* by the β parameters

Single-Hidden-Layer Feedforward Neural Network, cont’d



$$Y_i = g_Y(\beta_0 H_{i0} + \beta_1 H_{i1} + \cdots + \beta_{im-1} H_{im-1}) + \varepsilon_i$$

$$Y_i = g_Y(H_i'\beta) + \varepsilon_i$$

where:

$$\underset{m\times 1}{\boldsymbol{\beta}} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{m-1} \end{bmatrix} \qquad \underset{m\times 1}{\mathbf{H}_i} = \begin{bmatrix} H_{i0} \\ H_{i1} \\ \vdots \\ H_{i,m-1} \end{bmatrix}$$

$$H_{ij} = g_j(X_i'\alpha_j) \qquad \mathrm{j} = 1, \dots, \mathrm{m} - 1$$

Where $H_{i0} = 1\ and\ X_{i0} = 1$

$$\underset{p\times 1}{\boldsymbol{\alpha}_j} = \begin{bmatrix} \alpha_{j0} \\ \alpha_{j1} \\ \vdots \\ \alpha_{j,p-1} \end{bmatrix} \qquad \underset{p\times 1}{\mathbf{X}_i} = \begin{bmatrix} X_{i0} \\ X_{i1} \\ \vdots \\ X_{i,p-1} \end{bmatrix}$$

$$\Rightarrow \qquad Y_i = g_Y(\mathbf{H}_i'\boldsymbol{\beta}) + \varepsilon_i = g_Y\left[\beta_0 + \sum_{i=1}^{m-1} \beta_j g_i(\mathbf{X}_i'\boldsymbol{\alpha}_j)\right] + \varepsilon_i$$

Single-Hidden-Layer Feedforward Neural Network, cont’d

$$Y_i = g_Y(\mathbf{H}_i'\boldsymbol{\beta}) + \varepsilon_i = g_Y\left[\beta_0 + \sum_{i=1}^{m-1} \beta_j g_i(\mathbf{X}_i'\boldsymbol{\alpha}_j)\right] + \varepsilon_i$$

The m functions $\mathrm{g_Y}, \mathrm{g_1} \cdots, \mathrm{g_{m-1}}$ are called activation functions. A common choice for each of these functions is the logistic function

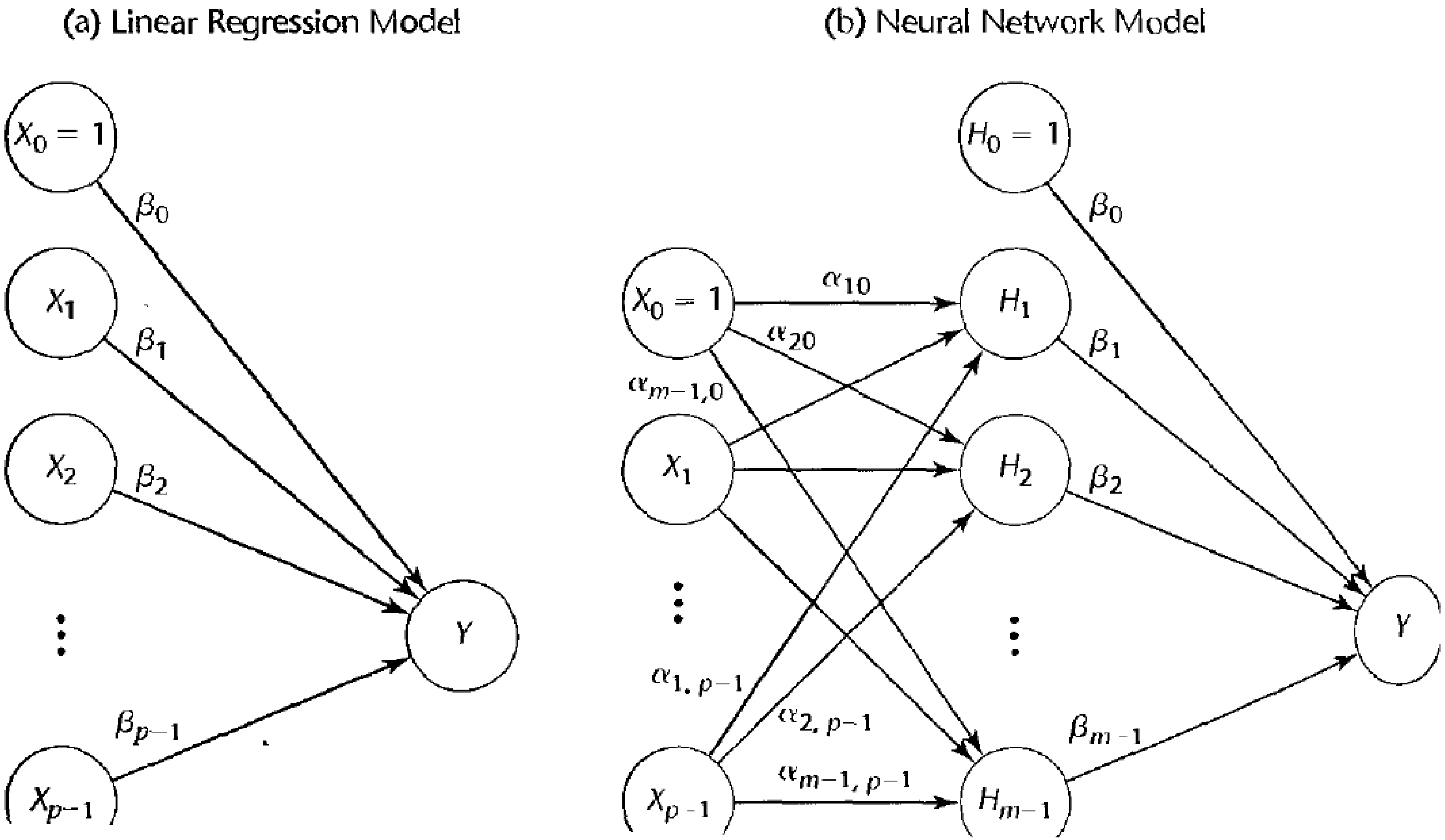
$$g(Z) = \frac{1}{1+e^{-Z}} = [1+e^{-Z}]^{-1}$$

$$g_i(\mathbf{X}_i'\boldsymbol{\alpha}_j) = [1 + \exp(-\alpha_{j0} - \alpha_{j1} X_{i1})]^{-1}$$

$$\begin{aligned} Y_i &= [1 + \exp(-\mathbf{H}_i'\boldsymbol{\beta})]^{-1} + \varepsilon_i \\ &= \left[1 + \exp\left[-\beta_0 - \sum_{j=1}^{m-1} \beta_i [1 + \exp(-\mathbf{X}_i'\boldsymbol{\alpha}_i)]^{-1}\right]\right]^{-1} + \varepsilon_i \\ &= f(\mathbf{X}_i, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{m-1}, \boldsymbol{\beta}) + \varepsilon_i \end{aligned}$$

Network Representation

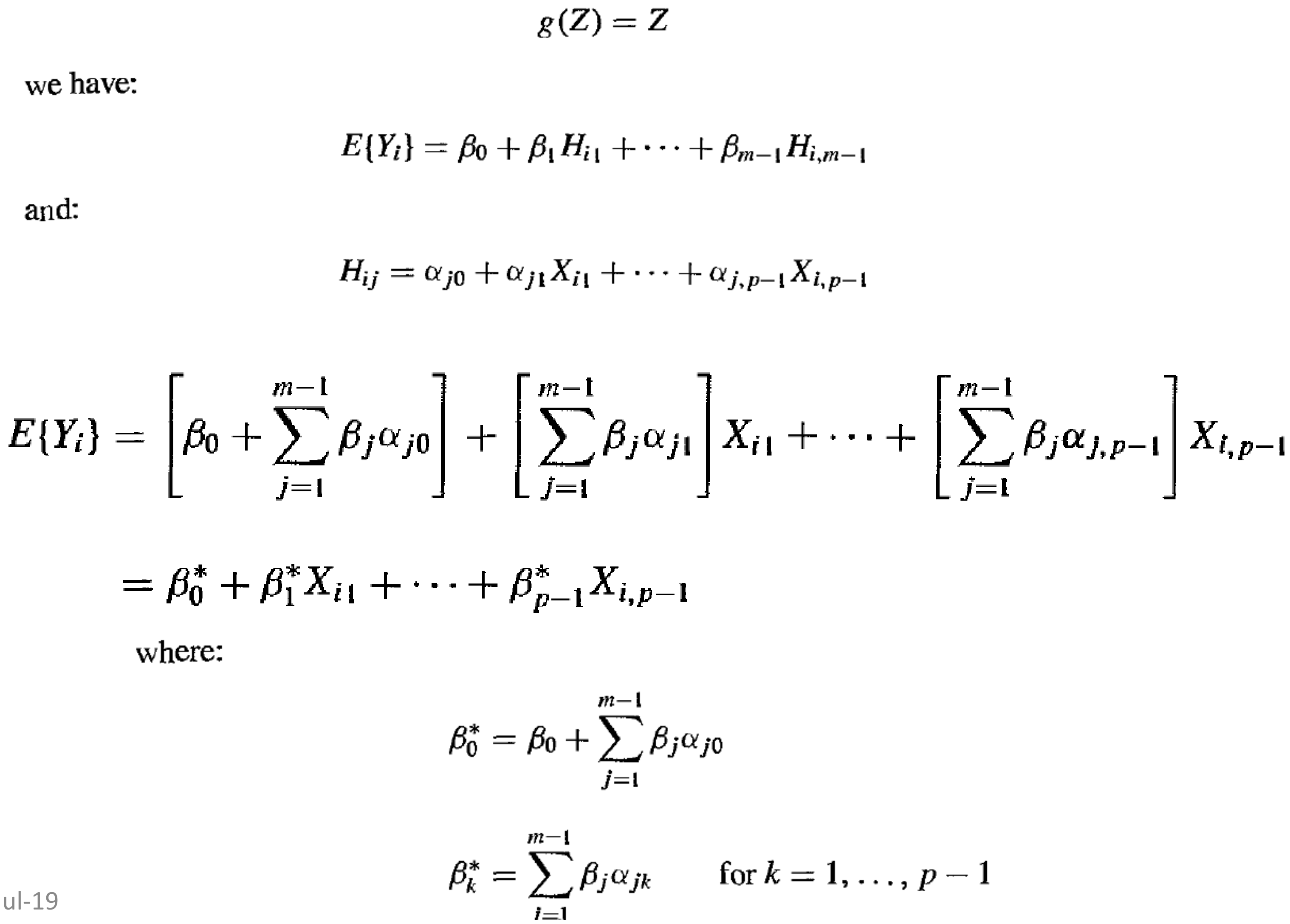
Linear regression is a special case of Neural Network models with identity activation functions


(a) Linear Regression Model

(b) Neural Network Model

Neural Network as Generalization of Linear Regression

Multiple Linear regression is a special case of Neural Network models with identity activation functions



$$g(Z) = Z$$

we have:

$$E\{Y_i\} = \beta_0 + \beta_1 H_{i1} + \cdots + \beta_{m-1} H_{i,m-1}$$

and:

$$H_{ij} = \alpha_{j0} + \alpha_{j1} X_{i1} + \cdots + \alpha_{j,p-1} X_{i,p-1}$$

$$E\{Y_i\} = \left[\beta_0 + \sum_{j=1}^{m-1} \beta_j \alpha_{j0}\right] + \left[\sum_{j=1}^{m-1} \beta_j \alpha_{j1}\right] X_{i1} + \cdots + \left[\sum_{j=1}^{m-1} \beta_j \alpha_{j,p-1}\right] X_{i,p-1}$$

$$= \beta_0^* + \beta_1^* X_{i1} + \cdots + \beta_{p-1}^* X_{i,p-1}$$

where:

$$\beta_0^* = \beta_0 + \sum_{j=1}^{m-1} \beta_j \alpha_{j0}$$

$$\beta_k^* = \sum_{i=1}^{m-1} \beta_j \alpha_{jk} \qquad \text{for } k = 1, \ldots, p-1$$

Parameter Estimation: Penalized Least Squares

The penalized least squares criterion is given by:

$$Q = \sum_{i=1}^{n} [Y_i - f(X_i, \beta, \alpha_1, \cdots, \alpha_{m-1})]^2 + p_\lambda(\beta, \alpha_1, \cdots, \alpha_{m-1})$$

where the overfit penalty is:

$$p_\lambda(\beta, \alpha_1, \cdots, \alpha_{m-1}) = \lambda \left[\sum_{i=0}^{m-1} \beta_i^2 + \sum_{i=1}^{m-1} \sum_{j=0}^{p-1} \alpha_{ij}^2 \right]$$

The penalty is a positive constant, λ, times the sum of squares of the nonlinear regression coefficients.

- the penalty is imposed not on the number of parameters m +mp, but on the total magnitude of the parameters.
- λ assigned to the regression coefficients governs the trade-off between overfitting and underfitting.

Parameter Estimation: Penalized Least Squares, cont'd

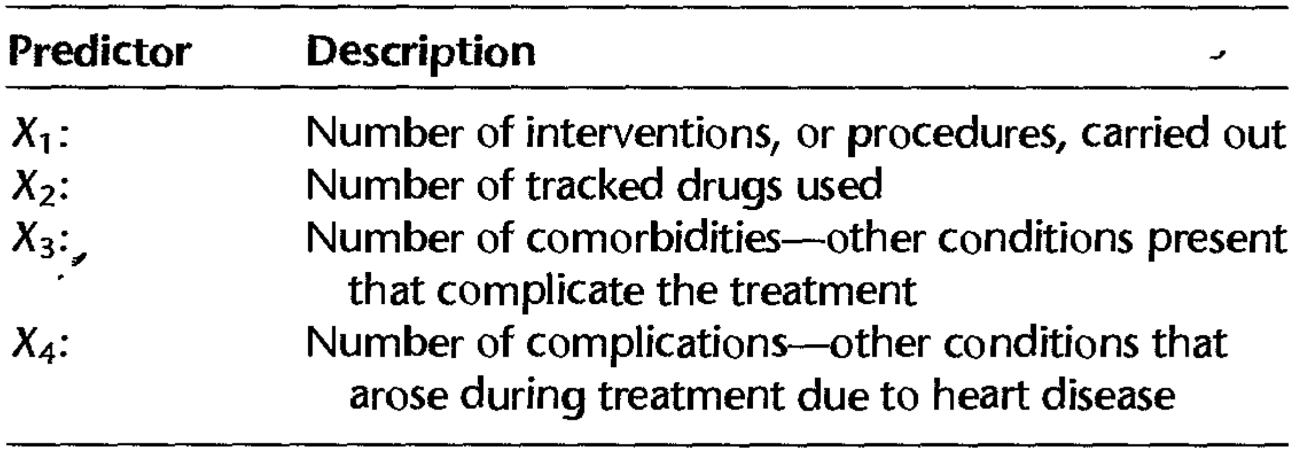
λ is large ⇒ the parameters estimates will be relatively small in absolute magnitude

λ is small ⇒ the parameters estimates will be relatively large

- There are many methods to estimate the coefficients!
- A "best" value for λ is generally between 0.001 and 0.1 and is chosen by cross-validation.
- Fit the model many times using different sets of randomly chosen starting values for each fit. This is referred to training the network.

Example: Ischemic Heart Disease

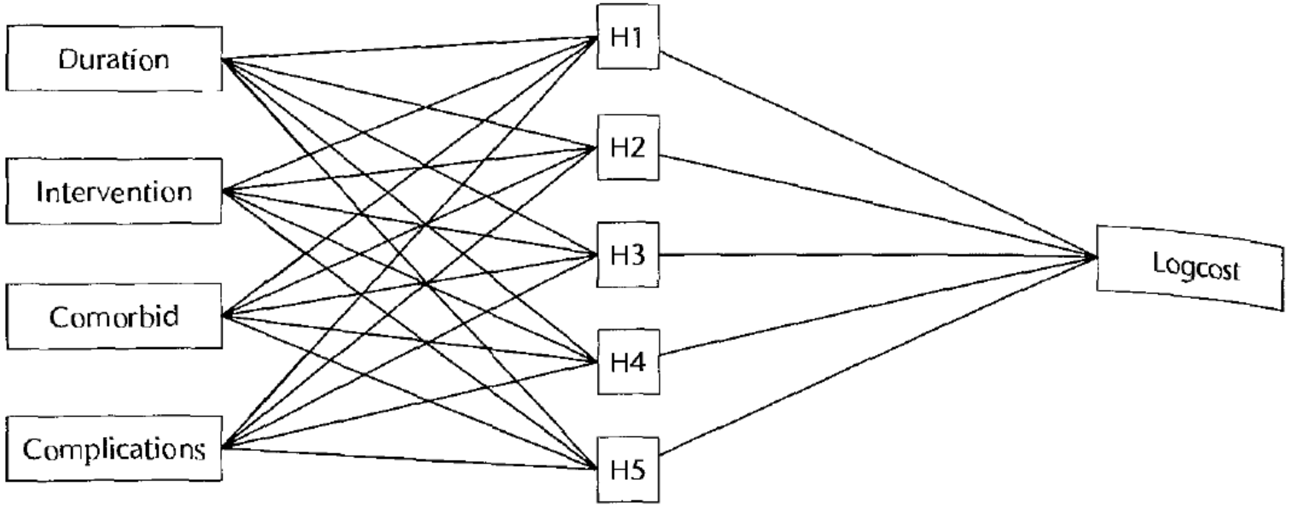
788 subscribers who made claims resulting from coronary heart disease.
The response *(Y)* is the logarithm of the total cost of services provided. The predictors to be studied here are:

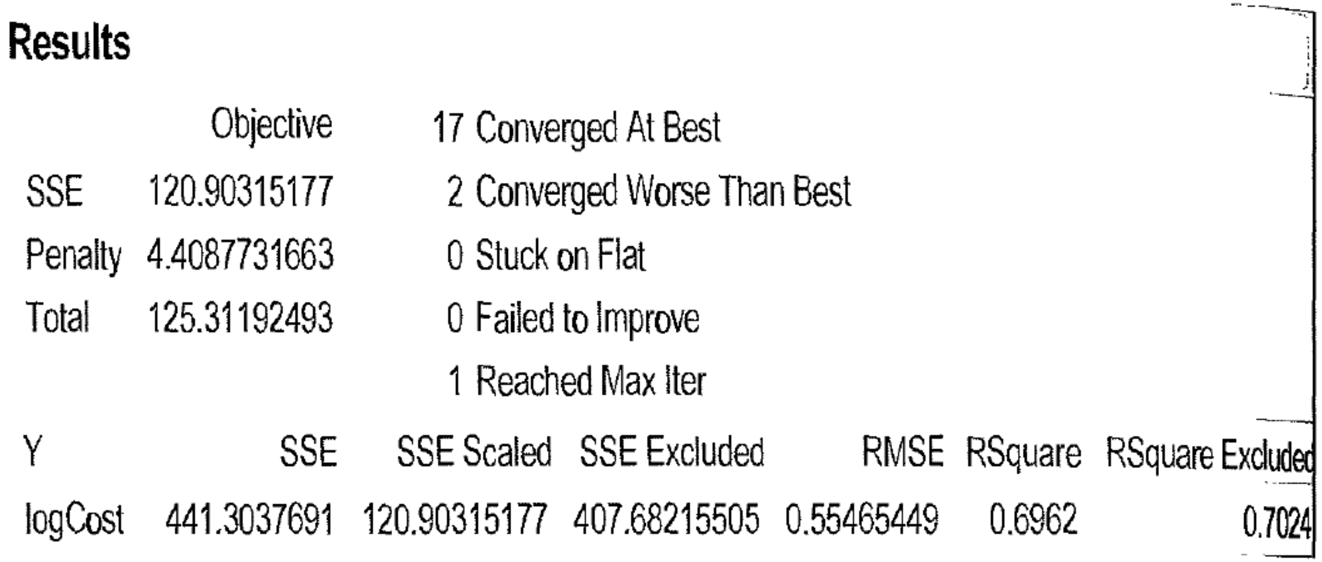


Predictor	Description
X_1:	Number of interventions, or procedures, carried out
X_2:	Number of tracked drugs used
X_3:	Number of comorbidities—other conditions present that complicate the treatment
X_4:	Number of complications—other conditions that arose during treatment due to heart disease

The first 400 observations are used to fit model (13.45)and the last n* = 388 observations were held out for validation. (Note that the observations were originally sorted in a random order, so that the hold-out data set is a random sample.)

Example: Ischemic Heart Disease, cont'd





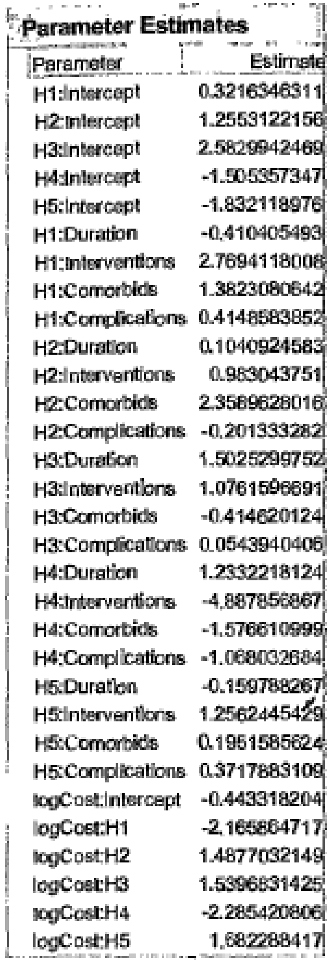
Results

```
            Objective       17 Converged At Best
SSE       120.90315177       2 Converged Worse Than Best
Penalty   4.4087731663       0 Stuck on Flat
Total     125.31192493       0 Failed to Improve
                             1 Reached Max Iter

Y                SSE   SSE Scaled   SSE Excluded        RMSE  RSquare  RSquare Excluded
logCost  441.3037691 120.90315177  407.68215505  0.55465449    0.6962            0.7024
```

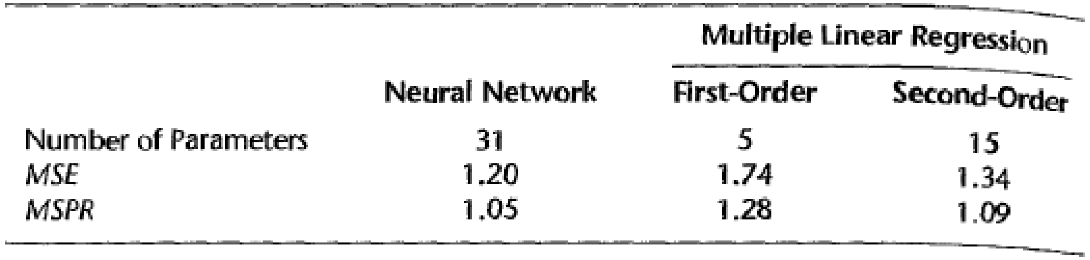
$$SSE_{VAL} = \sum_{i=401}^{788} (Y_i - \hat{Y}_i)^2 = 407.68$$

Example: Ischemic Heart Disease, cont’d

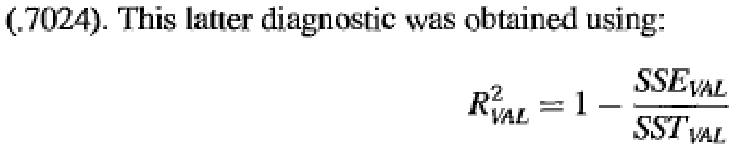


Parameter Estimates

Parameter	Estimate
H1:Intercept	0.3216346311
H2:Intercept	1.2553122156
H3:Intercept	2.5829942469
H4:Intercept	-1.505357347
H5:Intercept	-1.832118976
H1:Duration	-0.410405493
H1:Interventions	2.7694118008
H1:Comorbids	1.3823080642
H1:Complications	0.4148583852
H2:Duration	0.1040924583
H2:Interventions	0.983043751
H2:Comorbids	2.3589628016
H2:Complications	-0.201333282
H3:Duration	1.5025299752
H3:Interventions	1.0761596691
H3:Comorbids	-0.414620124
H3:Complications	0.0543940406
H4:Duration	1.2332218124
H4:Interventions	-4.887856867
H4:Comorbids	-1.576610999
H4:Complications	-1.068032684
H5:Duration	-0.159788267
H5:Interventions	1.2562445429
H5:Comorbids	0.1951585624
H5:Complications	0.3717883109
logCost:Intercept	-0.443318204
logCost:H1	-2.165864717
logCost:H2	1.4877032149
logCost:H3	1.5396831425
logCost:H4	-2.285420806
logCost:H5	1.682288417



		Multiple Linear Regression	
	Neural Network	First-Order	Second-Order
Number of Parameters	31	5	15
MSE	1.20	1.74	1.34
MSPR	1.05	1.28	1.09



(.7024). This latter diagnostic was obtained using:

$$R^2_{VAL} = 1 - \frac{SSE_{VAL}}{SST_{VAL}}$$

R code

- # install library
- install.packages("neuralnet ")
- # load library
- library(neuralnet)
- pop<-data.frame(cbind(y,x1,x2,x3,x4))
- data<-pop
- datatrain = data[1:400,]
- datatest = data[-c(1:400),]
- max = apply(pop , 2 , max)
- min = apply(pop, 2 , min)
- scaled = as.data.frame(scale(pop, center = min, scale = max - min))
- trainNN = scaled[1:400,]
- testNN = scaled[-c(1:400),]
- set.seed(123)
- NN = neuralnet(y ~ x1 + x2 + x3 + x4, trainNN, hidden = 5 , linear.output = T)

```
## Prediction using neural network

predict_testNN = compute(NN, testNN[,c(2:5)])
predict_testNN1 = (predict_testNN$net.result * (max(pop$y) - min(pop$y))) + min(pop$y)

plot(datatest$y, predict_testNN1, col='blue', pch=16, ylab = "predicted Y NN", xlab = "Actual Y")

abline(0,1)
```