Regression 11

Weighted least squares regression and Robust regression

1 Weighted least squares regression

The first assumption of the least squares regression we have studied is that $\mathrm{Var}(\epsilon_i) = \sigma^2$ for all cases in the data. This assumption is in doubt in many problems, as variances can depend on the response, on one or more of the predictors, or possibly on other factors.

If nonconstant variance is diagnosed, but exact variances are unknown, we could consider two remedies. First, a transformation of the response Y can be used. The second alternative is weighted least squares (WLS) with empirically chosen weights. Weights that are simple functions such as $\sigma_i^2 = \mathrm{Var}(\epsilon_i) = \sigma^2 X_{i1}$ are used. If large samples with replication are available, then within-group variances may be used to provide approximate weights. Generally, however, empirical weights that are functions of the $\hat{Y}_i$ or $\hat{\epsilon}_i$ from ordinary least squares (OLS) cannot be recommended unless nonstandard methods are used to estimate variances.

1.1 Parameter estimation by weighted least squares

Formerly, we assumed

$$\mathbf{Y} \ \sim \ N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}).$$

That is, the regression errors $\epsilon_1, \ldots, \epsilon_n$ were assumed to be *iid* $N(0, \sigma^2)$. Now suppose that the errors have unequal variances, which are known up to a proportionality constant,

$$\sigma_i^2 = \mathrm{Var}(\epsilon_i) = v_i\sigma^2, \qquad i = 1, \ldots, n,$$

where $v_1, \ldots, v_n$ are known. In matrix notation, we denote

$$\mathbf{Y} \ \sim \ N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{V}),$$

where $\mathbf{V} = \mathrm{diag}[v_1, \ldots, v_n]$. Hence we have

$$\boldsymbol{\epsilon} = \mathbf{Y} - \mathbf{X}\boldsymbol{\beta} \ \sim \ N(0, \sigma^2\mathbf{V}).$$

Let us denote $\mathbf{W}^{1/2} = \mathrm{diag}\big(1/\sqrt{v_1}, \ldots, 1/\sqrt{v_n}\big)$. Notice that $\mathbf{W} = \mathbf{W}^{1/2}\mathbf{W}^{1/2} = \mathbf{V}^{-1}$. Then we have

$$\mathbf{W}^{1/2}\boldsymbol{\epsilon} = \mathbf{W}^{1/2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \ \sim \ N(0, \sigma^2\mathbf{I}).$$

For convenience, let us denote

$$\boldsymbol{\epsilon}^* = \mathbf{W}^{1/2}\boldsymbol{\epsilon}, \qquad \mathbf{Y}^* = \mathbf{W}^{1/2}\mathbf{Y}, \quad \text{and} \quad \mathbf{X}^* = \mathbf{W}^{1/2}\mathbf{X}.$$

Thus, the weighted least squares (WLS) is equivalent to the OLS estimator on $\mathbf{Y}^*$ and $\mathbf{X}^*$:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= (\mathbf{X}^{*\prime}\mathbf{X}^*)^{-1}\mathbf{X}^{*\prime}\mathbf{Y}^* = (\mathbf{X}'\mathbf{W}^{1/2}\mathbf{W}^{1/2}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{1/2}\mathbf{W}^{1/2}\mathbf{Y} \\ &= (\mathbf{X}'\mathbf{W}\mathbf{Y})^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}.\end{aligned}$$

That is, the WLS estimator is equivalent to minimizing

$$Q_W = \|\boldsymbol{\epsilon}^*\|^2 = \sum_{i=1}^{n} w_i \cdot \big\{Y_i - (\beta_0 + \beta_1 X_i + \ldots + \beta_{p-1}X_{p-1})\big\}^2. \tag{11.1}$$

For convenience, we define the row vectors in the data matrix $\mathbf{X}$ by $\mathbf{x}_i{}' = [1\ X_{i1}\ X_{i2}\ \cdots\ X_{i,p-1}]$ so that we have

$$\underset{n\times p}{\mathbf{X}} = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1,p-1} \\ 1 & X_{21} & X_{22} & \cdots & X_{2,p-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{n1} & X_{n2} & \cdots & X_{n,p-1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1{}' \\ \mathbf{x}_2{}' \\ \vdots \\ \mathbf{x}_n{}' \end{bmatrix}.$$

Then we can rewrite (11.1) as

$$Q_W = \sum_{i=1}^{n} w_i \cdot \left\{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\right\}^2. \tag{11.2}$$

Note that if $w_i = 1$ for all i, then this is equivalent to the OLS.

Differentiating (11.2) with respect to $\boldsymbol{\beta}$, we have

$$\frac{\partial Q_W}{\partial \boldsymbol{\beta}} = 2\sum_{i=1}^{n} w_i \cdot \left\{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\right\}\mathbf{x}_i{}'.$$

Thus, the WLS estimator is also obtained by solving

$$\sum_{i=1}^{n} w_i \cdot \left\{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\right\}\mathbf{x}_i{}' = \mathbf{0}, \tag{11.3}$$

where $\mathbf{0} = (0, 0, \ldots, 0)$.

1.2 Where do we get the weights

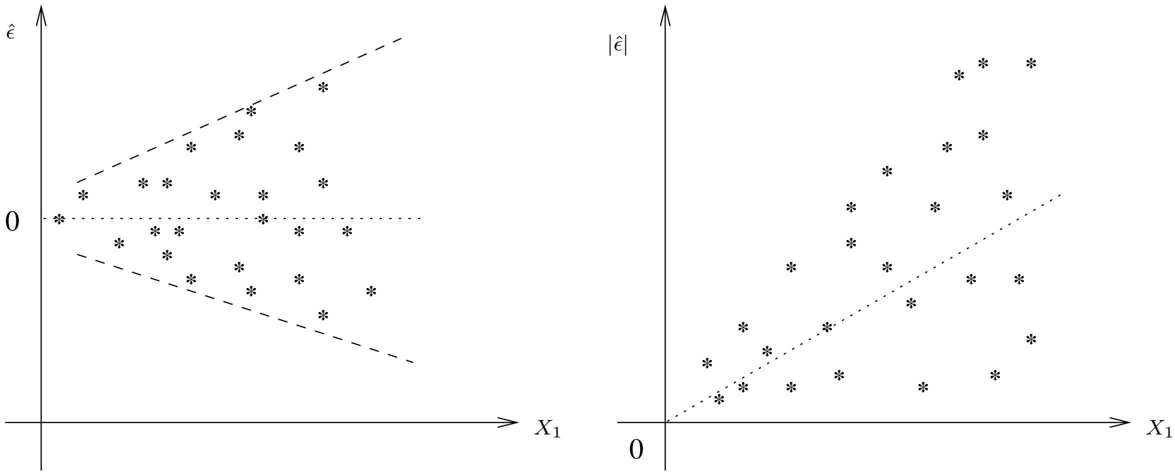
In WLS, we assume that $\sigma_i^2 = \mathrm{Var}(\epsilon_i) = \sigma^2 v_i$, where $v_1, \ldots, v_n$ are known. Where do we get the weights in real data analysis? Note that the weights is inversely proportional to the variance σ_i^2, that is, $w_i \propto 1/\sigma_i^2$, or, $w_i \propto 1/v_i$ in this case.

1. From a prediction variable.

   Suppose that we fit an OLS regression, and a residual plot of $\hat{\epsilon}$ versus a predictor X_1 looks like this.

Residual plot against X_1

Residual plot against X_1



Then we might suppose that

$$\mathrm{Var}(\epsilon) = \sigma^2 X_1^{1/2}$$

$$\mathrm{Var}(\epsilon) = \sigma^2 X_1$$

$$\mathrm{Var}(\epsilon) = \sigma^2 X_1^2$$

$$\vdots$$

$$\mathrm{Var}(\epsilon) = \sigma^2(\hat{\alpha}_0 + \hat{\alpha}_1 X_1)^2,$$

where $\hat{\alpha}_0$ and $\hat{\alpha}_1$ are obtained by regressing $|\hat{\epsilon}|$ on X_1.

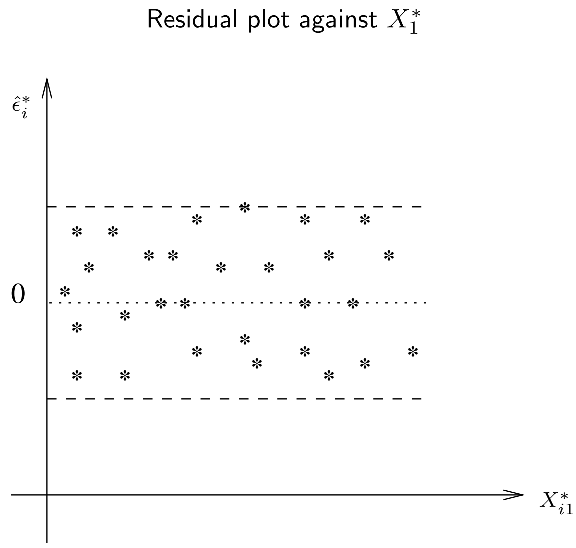
How do we know which power of X_1 to use? Suppose, for example, we try $\mathrm{Var}(\epsilon) = \sigma^2 X_1^{1/2}$, *i.e.*, we use $w_i = 1/X_{i1}^{1/2}$ as the weights. If the residual plot from this regression ($\hat{\epsilon}_i^* = w_i^{1/2}(Y_i - \hat{Y}_i)$ versus $X_{i1}^* = w_i^{1/2} X_{i1}$) looks good, then our variance function $\mathrm{Var}(\epsilon) = \sigma^2 X_1^{1/2}$ is OK. If the plot still fans out, then we need to use a stronger variance function (*e.g.*, $\mathrm{Var}(\epsilon) = \sigma^2 X_1$ or $\mathrm{Var}(\epsilon) = \sigma^2 X_1^2$).

2. From replication

Suppose that only a few distinct patterns of predictors are present. For example,

$$X = \begin{cases} 0 & : \quad \text{female} \\ 1 & : \quad \text{male} \end{cases}$$

Residual plot against X_1^*



If the sample sizes within each group are large enough, we can estimate σ^2 within each group.

3. From varying sample sizes

   Suppose that our responses Y_i are actually averages from sample of varying sizes. For example, let Y_i be the average wage for workers at the ith firm and n_i be the number of workers at the ith firm. Then we expect Y_i to have more random variation when n_i is smaller than when n_i is large. In building a regression model for $Y_1, \ldots, Y_n$, it may be sensible to assume that $\mathrm{Var}(\epsilon_i) \propto 1/n_i$ and thus use the n_i's as weights.

Example 1. Textbook Example (Table 11.1 on Page 427).

`Minitab`

Read Data

```
MTB >READ c1 c2;
SUBC>  file  "S:\LM\CH11TA01.TXT" .
Entering data from file: S:\LM\CH11TA01.TXT
54 rows read.
```

Regression of C2 on C1

```
MTB > regr c2 1 c1;     # c2 = blood pressure    c1 = age
SUBC> resid c3;
SUBC> fits  c4;
SUBC> brief 1.

```

```
Regression Analysis: C2 versus C1
The regression equation is
C2 = 56.2 + 0.580 C1
Predictor      Coef   SE Coef      T      P
Constant     56.157     3.994  14.06  0.000
C1          0.58003   0.09695   5.98  0.000

S = 8.14575   R-Sq = 40.8%   R-Sq(adj) = 39.6%

Analysis of Variance
Source          DF      SS      MS      F      P
Regression       1  2375.0  2375.0  35.79  0.000
Residual Error  52  3450.4    66.4
Total           53  5825.3

Residual Plots for C2
```

Regression of C5 ($|\hat{\epsilon}_i|$) on C1

```
MTB > let c5 = abs(c3)
MTB > regr c5 1 c1;
SUBC> fits c6;
SUBC> brief 1.

Regression Analysis: C5 versus C1

The regression equation is
C5 = - 1.55 + 0.198 C1

Predictor      Coef   SE Coef      T      P
Constant     -1.549     2.187  -0.71  0.482
C1          0.19817   0.05309   3.73  0.000

S = 4.46057   R-Sq = 21.1%   R-Sq(adj) = 19.6%

Analysis of Variance

Source          DF       SS      MS      F      P
Regression       1   277.23  277.23  13.93  0.000
Residual Error  52  1034.63   19.90
Total           53  1311.86

Residual Plots for C5
```

Table 11.1 on Page 427

```
MTB > let c7 = 1/c6      # c7 = weight^(1/2)
MTB > let c8 = c7*c7     # c8 = weight
MTB > print c1 c2 c3 c5 c6 c8

Data Display
Row  C1   C2       C3       C5       C6        C8
  1  27   73   1.1822   1.1822   3.8012  0.069209
  2  21   66  -2.3376   2.3376   2.6121  0.146557
  3  22   63  -5.9176   5.9176   2.8103  0.126617
  4  24   75   4.9223   4.9223   3.2067  0.097251
 .................................................

 51  50   91   5.8415   5.8415   8.3591  0.014311
 52  52  100  13.6815  13.6815   8.7555  0.013045
 53  58   80  -9.7987   9.7987   9.9445  0.010112
 54  57  109  19.7813  19.7813   9.7463  0.010527
```

WLS (using weight option)

```
MTB > regr c2 1 c1;
SUBC> weight c8;
SUBC> resid  c9;
SUBC> fits   c10;
SUBC> brief 1.
```

```

Regression Analysis: C2 versus C1

Weighted analysis using weights in C8

The regression equation is
C2 = 55.6 + 0.596 C1

Predictor      Coef   SE Coef      T      P
Constant     55.566     2.521  22.04  0.000
C1          0.59634   0.07924   7.53  0.000

S = 1.21302   R-Sq = 52.1%   R-Sq(adj) = 51.2%


Analysis of Variance
Source          DF       SS      MS      F      P
Regression       1   83.341  83.341  56.64  0.000
Residual Error  52   76.514   1.471
Total           53  159.854

Residual Plots for C2
```

WLS (using OLS)

```
MTB > let c12 = c7*c1    # c12 = X*
MTB > let c22 = c7*c2    # c22 = Y*
MTB > regr c22 2 c7 c12 ;  # Regress Y* on X*
SUBC> noconstant ;
SUBC> fits c23;
SUBC> brief 1.

Regression Analysis: C22 versus C7, C12

The regression equation is
C22 = 55.6 C7 + 0.596 C12

Predictor      Coef   SE Coef      T      P
Noconstant
C7           55.566     2.521  22.04  0.000
C12         0.59634   0.07924   7.53  0.000
S = 1.21302

Analysis of Variance
Source          DF       SS      MS        F      P
Regression       2  12446.6  6223.3  4229.48  0.000
Residual Error  52     76.5     1.5
Total           54  12523.1
Residual Plots for C22
```

Print c10 and c23. Print c10 and c24

```
MTB > print c10 c23

Data Display
Row      C10      C23
  1  71.6670  18.8539
  2  68.0889  26.0663
  3  68.6853  24.4404
  4  69.8780  21.7915
     ................
 51  85.3829  10.2143
 52  86.5755   9.8882
 53  90.1536   9.0657
 54  89.5572   9.1888

MTB > let c24 = c23/c7
MTB > print c10 c24

Data Display
Row      C10      C24
```

```
  1  71.6670  71.6670
  2  68.0889  68.0889
  3  68.6853  68.6853
  4  69.8780  69.8780
     ................
 51  85.3829  85.3829
 52  86.5755  86.5755
 53  90.1536  90.1536
 54  89.5572  89.5572
```

R

Read Data

```
> mydata =
    read.table("https://raw.githubusercontent.com/AppliedStat/LM/master/CH11TA01.txt")
> x = mydata[,1]  ;  y = mydata[,2]
```

Regression of y (blood pressure) on x (age)

```
> LM = lm ( y ~ x )
> summary(LM)
Call:
lm(formula = y ~ x)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 56.15693    3.99367  14.061  < 2e-16 ***
x            0.58003    0.09695   5.983 2.05e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.146 on 52 degrees of freedom
Multiple R-Squared: 0.4077, Adjusted R-squared: 0.3963
F-statistic: 35.79 on 1 and 52 DF,  p-value: 2.050e-07
```

Regression of c5 ($|\hat{\epsilon}_i|$) on x (age)

```
> c3 = resid(LM)
> c4 = fitted(LM)
>
> c5 = abs (c3)
>
> LM2 = lm ( c5 ~ x )
> summary(LM2)

Call:
lm(formula = c5 ~ x)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.54948    2.18692  -0.709  0.48179
x            0.19817    0.05309   3.733  0.00047 ***
---
Residual standard error: 4.461 on 52 degrees of freedom
Multiple R-squared:  0.2113,    Adjusted R-squared:  0.1962
F-statistic: 13.93 on 1 and 52 DF,  p-value: 0.0004705
```

Table 11.1 on Page 427

```
> c6  = fitted (LM2)
> c7 = 1/c6        # c7 = w^(1/2)
> c8 = c7 * c7     # c8 = w

> cbind(x, y, c3, c5, c6, c8)
   x   y         c3         c5       c6          c8
1  27  73   1.1822391  1.1822391  3.801175 0.069209280
```

```
2  21  66  -2.3375761  2.3375761  2.612141 0.146557083
3  22  63  -5.9176069  5.9176069  2.810313 0.126616574
4  24  75   4.9223315  4.9223315  3.206658 0.097251155
.......................................................

51 50  91   5.8415308  5.8415308  8.359138 0.014311232
52 52 100  13.6814692 13.6814692  8.755482 0.013044872
53 58  80  -9.7987156  9.7987156  9.944516 0.010111898
54 57 109  19.7813152 19.7813152  9.746344 0.010527289
```

WLS using lm() with options

```
>  WLS = lm ( y ~ x , weights = c8)   # weights are given in c8

>  summary (WLS)
Call:
lm(formula = y ~ x, weights = c8)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 55.56577    2.52092  22.042  < 2e-16 ***
x            0.59634    0.07924   7.526 7.19e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.213 on 52 degrees of freedom
Multiple R-Squared: 0.5214, Adjusted R-squared: 0.5122
F-statistic: 56.64 on 1 and 52 DF,  p-value: 7.187e-10

>  c9  = resid (WLS)
>  c10 = fitted (WLS)
```

WLS using OLS, that is, lm() without options

```
> c12 = c7 * x  # c12 = X*
> c22 = c7 * y  # c22 = Y*

> LM2 = lm ( c22 ~ 0 + c7 + c12 )  # Regress Y* on X*
> c23 = fitted(LM2)
> summary(LM2)
Call:
lm(formula = c22 ~ 0 + c7 + c12)

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
c7   55.56577    2.52092  22.042  < 2e-16 ***
c12   0.59634    0.07924   7.526 7.19e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.213 on 52 degrees of freedom
Multiple R-Squared: 0.9939, Adjusted R-squared: 0.9937
F-statistic:  4229 on 2 and 52 DF,  p-value: < 2.2e-16
```

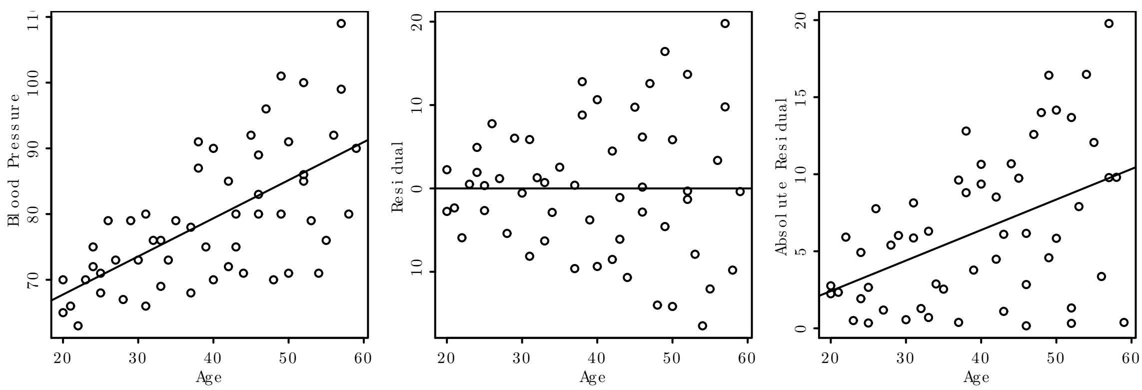
Print c10 and c23. Print c10 and c24

```
>  cbind(c10, c23)
        c10       c23
1  71.66699 18.853906
2  68.08894 26.066336
3  68.68528 24.440438
4  69.87797 21.791526
.......................................................

51 85.38285 10.214313
52 86.57554  9.888151
53 90.15359  9.065658
54 89.55724  9.188804

>  c24 = c23/c7
```

```
> cbind(c10, c24)
       c10      c24
1  71.66699 71.66699
2  68.08894 68.08894
3  68.68528 68.68528
4  69.87797 69.87797
..........................................................

51 85.38285 85.38285
52 86.57554 86.57554
53 90.15359 90.15359
54 89.55724 89.55724
```

Figure 11.1 on Page 428

```
> par (mfrow=c(1,3))
> plot(x,y)       # x=age, y=blood pressure
> abline(LM)

> plot(x,c3)      # c3=residual of blood pressures
> abline( lm(c3~x) )

> plot(x,c5)      # c5=abs(c3)
> abline( lm(c5~x) )
```



2 Robust Regression

We can also write the OLS as

$$Q_2 = \sum_{i=1}^{n} \{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\}^2 = \rho(Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}), \tag{11.4}$$

where $\rho(t) = t^2$.

Differentiating (11.4) with respect to $\boldsymbol{\beta}$, we have

$$\frac{\partial Q_2}{\partial \boldsymbol{\beta}} = 2\sum_{i=1}^{n} \psi(Y_i - \mathbf{x}_i{}'\boldsymbol{\beta})\mathbf{x}_i{}'$$

where $\psi(t) = \rho'(t)$. Thus, the OLS estimator is also obtained by solving

$$\sum_{i=1}^{n} \psi\big(Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\big)\mathbf{x}_i{}' = \mathbf{0}. \tag{11.5}$$

If $\psi(t)$ is Winsorized at c ($\psi_c(t) = t$ for $|x| \leq c$ and $\psi_c(t) = c$ for $|x| > c$), then we can obtain the robustness. We can choose $c = k\sigma$. It is known that $c = 1.345\sigma$ give 95% efficiency at the normal model. (95 is often used for a magic number in statistics). It should be noted that $\psi_c(t) = \psi_k(t/\sigma) \cdot \sigma$ where $c = k\sigma$. This ψ_k is also known as the Huber's ψ function with the tuning constant k. The equation (11.5) can be rewritten as

$$\sum_{i=1}^{n} \psi_c\big(Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}\big)\mathbf{x}_i{}' = \sum_{i=1}^{n} \frac{\psi_k\Big(\frac{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}}{\sigma}\Big)\sigma}{\frac{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}}{\sigma}} \Big(\frac{Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}}{\sigma}\Big)\mathbf{x}_i{}' = \mathbf{0}.$$

For convenience, let $u_i = (Y_i - \mathbf{x}_i{}'\boldsymbol{\beta})/\sigma$. Then we have

$$\sum_{i=1}^{n} \frac{\psi_k(u_i)}{u_i}(Y_i - \mathbf{x}_i{}'\boldsymbol{\beta})\mathbf{x}_i{}' = \sum_{i=1}^{n} w_k(u_i) \cdot (Y_i - \mathbf{x}_i{}'\boldsymbol{\beta})\mathbf{x}_i{}' = \mathbf{0},$$

which is in a form of the WLS. The weight function $w(t)$ is also known as the Huber weight function which is given by

$$w_k(u) = \begin{cases} 1 & \text{if} \quad u \leq k \\ \frac{k}{|u|} & \text{if} \quad u > k \end{cases}$$

The problem is how to find the weights $w_k(u_i)$ and solve the equation above. An iterative method (iteratively reweighted least squares, IRLS) can be applied. Let m be the m-th step in the iterative algorithm. Let $\boldsymbol{\beta}^{(m)}$ be the estimate of the parameter vector and $\hat{\sigma}^{(m)}$ be the scale estimate obtained at the m-th step. Denote $u_i^{(m)} = (Y_i - \mathbf{x}_i{}'\boldsymbol{\beta}^{(m)})/\hat{\sigma}^{(m)}$. Then the parameter vector $\boldsymbol{\beta}$ is estimate as follows.

1. Select initial estimate $\boldsymbol{\beta}^{(0)}$ and estimate $\hat{\sigma}^{(0)}$.

   The $\boldsymbol{\beta}^{(0)}$ is usually obtained using the OLS, and $\hat{\sigma}^{(0)}$ is usually obtained by the MAD of the residuals.

2. At the m-th iteration step, estimate $\boldsymbol{\beta}^{(m)}$ and $\hat{\sigma}^{(m)}$ using the WLS with the previous values ($\boldsymbol{\beta}^{(m-1)}$ and $\hat{\sigma}^{(m-1)}$).

This is, we solve the following for $\boldsymbol{\beta}$ and let the solution denote $\boldsymbol{\beta}^{(m)}$:

$$\sum_{i=1}^{n} w_k(u_i^{(m-1)}) \cdot (Y_i - \mathbf{x}_i{}'\boldsymbol{\beta})\mathbf{x}_i{}' = \mathbf{0}.$$

3. Repeat Steps 1 and 2.

Example 2. Textbook Example 1 on Page 441. The education testing service (ETS) study data set are provided. The mathematics proficiency (Y) is regressed on X_2 (home library) using the robust regression. Note that Figure 11.5 on Page 442 has a typo (X_3 in the figure should read X_2).

R

Read Data

```
>
  Data=read.table("https://raw.githubusercontent.com/AppliedStat/LM/master/CH11TA04.txt")
> y  = Data[,2]
> X2 = Data[,4]
```

OLS

```
> X2bar = mean(X2); x2 = X2 - X2bar  # de-meaned

> # OLS
> LM0 = lm( y~x2 + I(x2^2) )
> e0 = resid(LM0)
> u0 = e0 / mad(e0)
```

WLS with Huber

```
> weight.huber  <- function(x, k=1.345) { pmin(1, k/abs(x) ) }

> # WLS: 1st iteration
> w1 = weight.huber(u0)
> LM1 = lm( y~x2 + I(x2^2), weights=w1)
> e1 = resid(LM1)

> # WLS: 2nd iteration
> u1= e1 / mad(e1)
> w2 = weight.huber(u1)
> LM2 = lm( y~x2 + I(x2^2), weights=w2)
> e2 = resid(LM2)

> # WLS: 3rd iteration
> u2= e2 / mad(e2)
> w3 = weight.huber(u2)
> LM3 = lm( y~x2 + I(x2^2), weights=w3)
> e3 = resid(LM3)

> # WLS: 4th iteration
> u3= e3 / mad(e3)
> w4 = weight.huber(u3)
> LM4 = lm( y~x2 + I(x2^2), weights=w4)
> e4 = resid(LM4)

> # WLS: 5th iteration
> u4= e4 / mad(e4)
> w5 = weight.huber(u4)
> LM5 = lm( y~x2 + I(x2^2), weights=w5)
```

```
> e5 = resid(LM5)

> # WLS: 6th iteration
> u5= e5 / mad(e5)
> w6 = weight.huber(u5)
> LM6 = lm( y~x2 + I(x2^2), weights=w6)
> e6 = resid(LM6)

> # WLS: 7th iteration
> u6= e6 / mad(e6)
> w7 = weight.huber(u6)
> LM7 = lm( y~x2 + I(x2^2), weights=w7)
> e7 = resid(LM7)
```

Iteratively Huber-Reweighted least squares

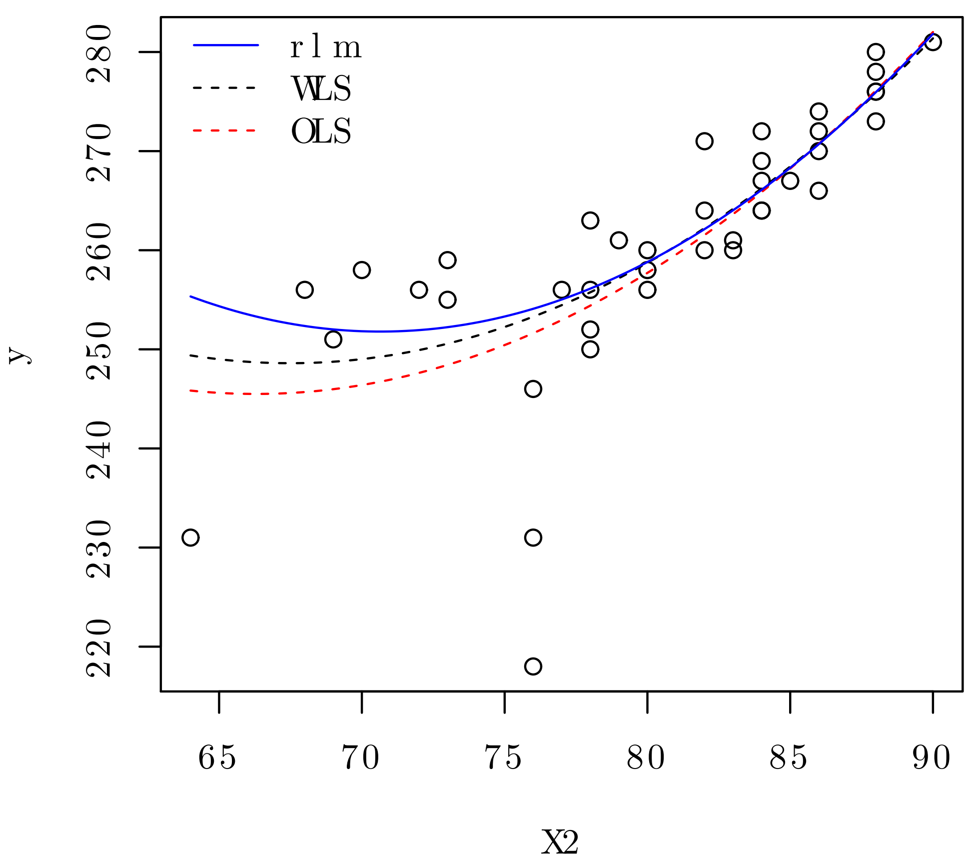
```

> # Table 11.5 (Page 444)
> round(cbind(e0,u0, w1,e1, w2, e2, w7,e7),4)
         e0      u0     w1       e1     w2       e2     w7       e7
1    -2.4109 -0.5164 1.0000   -3.7542 1.0000   -4.0354 1.0000   -4.1269
2    10.5724  2.2646 0.5939    8.4297 0.7152    7.4848 0.8601    6.7698
3     3.0454  0.6523 1.0000    1.5411 1.0000    1.1559 1.0000    0.9731
4    10.3104  2.2085 0.6090    7.3822 0.8166    5.4138 1.0000    3.6583
    ..............................
8   -20.6282 -4.4186 0.3044  -22.2929 0.2704  -22.7964 0.2526  -23.0873
    ..............................
11  -14.8358 -3.1779 0.4232  -18.3824 0.3280  -21.4286 0.2402  -24.3167
    ..............................
36  -33.6282 -7.2032 0.1867  -35.2929 0.1708  -35.7964 0.1616  -36.0873
37    2.4659  0.5282 1.0000    1.7722 1.0000    1.7627 1.0000    1.8699
38   -1.7129 -0.3669 1.0000   -2.7325 1.0000   -2.8491 1.0000   -2.8079
39    3.2658  0.6995 1.0000    3.2304 1.0000    3.2624 1.0000    3.3014
40    1.2658  0.2711 1.0000    1.2304 1.0000    1.2624 1.0000    1.3014
```

Using rlm() MASS library, which is slightly different from the above

```

> library(MASS)
> RLM1 = rlm(y~x2 + I(x2^2), method="M", scale.est="MAD", k2=1.345, maxit=1 )
> RLM1
Call:
rlm(formula = y ~ x2 + I(x2^2), scale.est = "MAD", k2 = 1.345,
    maxit = 1, method = "M")
Ran 1 iterations without convergence
Coefficients:
 (Intercept)           x2       I(x2^2)
259.38160409   1.67081807   0.06476101

> RLM7 = rlm(y~x2 + I(x2^2), method="M", scale.est="MAD", k2=1.345, maxit=7 )
> RLM7
Call:
rlm(formula = y ~ x2 + I(x2^2), scale.est = "MAD", k2 = 1.345,
    maxit = 7, method = "M")
Ran 7 iterations without convergence
Coefficients:
 (Intercept)           x2       I(x2^2)
259.42100205   1.56491518   0.08016681

> RLM  = rlm(y~x2 + I(x2^2))
> RLM
Call:
rlm(formula = y ~ x2 + I(x2^2))
Converged in 10 iterations
Coefficients:
 (Intercept)           x2       I(x2^2)
259.42112605   1.56460704   0.08021299
```

Plot to compare OLS, WLS and rlm()

```
> # Scatter plot and fitted curves
> postscript(file="Figure-11-5.eps", width=5, height=5)
> plot(X2, y)
> legend(63,285, bty="n", lty=c(1,2,2), col=c("blue","black","red"),
+         legend=c("rlm", "WLS", "OLS") )
> # From LM0 (OLS)
> curve( 258.43557+1.83272*(x-X2bar)+0.06491*(x-X2bar)^2,add=TRUE, col="red", lty=2)
>
> # From LM1 (WLS after 1st iteration)
> curve( 259.39021+ 1.67011*(x-X2bar)+0.06463*(x-X2bar)^2,add=TRUE, lty=2)
>
> # From RLM (MASS library) # same as WLS after 10th iteration.;
> curve( 259.421126+ 1.5646*(x-X2bar)+0.08021299*(x-X2bar)^2,add=TRUE, col="blue")
```



Remark 1. There are other ψ functions which give M-estimators (MLE-like estimators).

- Metric trimming.
- Metric Winsorizing (also called Huber).
- Tukey's biweight.
- Hampel's ψ.

△

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computers & Industrial Engineering 48 (2005) 173–180

computers & industrial engineering

www.elsevier.com/locate/dsw

Robust design modeling and optimization with unbalanced data

Byung Rae Cho[a,*], Chanseok Park[b]

[a]*Advanced Quality Engineering Laboratory, Department of Industrial Engineering, Clemson University, Clemson, SC 29634-0920, USA*
[b]*Department of Mathematical Sciences, Clemson University, Clemson, SC 29634, USA*

Abstract

The usual assumption behind robust design is that the number of replicates at each design point during an experimental stage is equal. In practice, however, it is often the case that this assumption is not met due to physical limitations and/or cost constraints. In this situation, using the usual method of ordinary least squares (OLS) to obtain fitted response functions for the mean and variance of the quality characteristic of interest may not be an effective tool. In this paper, we first show simulation results, indicating that an alternative method, called the method of weighted least squares (WLS), outperforms the OLS method in terms of mean squared error. We then lay out the WLS-based robust design modeling and optimization. A case study is presented for numerical purposes.

Keywords: Quality improvement; Robust design; Weighted least squares; Simulation; Optimization

1. Introduction

Among the engineering design methods for the purpose of quality improvement which are currently studied in the research community and implemented in industries, robust design (RD) is often identified as one of the most attractive methodologies. In fact, major US industries have promoted and implemented RD techniques to improve product quality significantly. Many applications can be found in Bendell, Disney, and Pridmore (1987) and Dehnad (1989) for engineering problems in the automotive industry, plastic technology, process industry, and information technology.

The ultimate objective of RD is to minimize variation in the quality characteristic of interest, while keeping a process mean at the customer-identified target value. While the basic concept underlying RD is clearly important, Taguchi's tools for achieving this goal, such as orthogonal arrays and signal-to-noise

* Corresponding author. Tel.: +1 864 656 1874; fax: +1 864 656 0795.
E-mail address: bcho@ces.clemson.edu (B.R. Cho).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2005.01.004

ratios, have drawn much criticism. Initiated by Box (1985), several authors including Myers and Montgomery (2002); Myers, Khuri, and Vining (1992); Pignatiello and Ramberg (1991); Tiao, Bisgaard, Hill, Peña, and Stigler (2000); and Vining and Myers (1990) pointed out shortcomings embodied in Taguchi's approach to RD. Consequently, there has been a great deal of research effort to rectify these drawbacks. One of the alternatives, the dual response approach using response surface methodology, has received much attention (see Vining & Myers, 1990). This approach facilitates the understanding of the system by separately modeling the response functions for process mean and variance. The dual response approach has been further studied by several researchers, including Cho, Kim and Cho (2000, 2002); Del Castillo and Montgomery (1993); Lin and Tu (1995); and Kim, Kimbler, and Phillips (2000).

2. Research motivation

Most dual response models in the literature have assumed the same number of replicates at each design point during an experimental stage. This assumption, however, does not hold in every modeling application. There are some practical situations in which different sample sizes are considered. Examples of these unbalanced designs are numerous.

(1) A quality engineer may have designed an experiment that would yield an equal number of replicates at each design point. However, unforeseen problems may occur that result in the loss of some observations. Thus the engineer ends up with unbalanced data.
(2) Certain treatment combinations may be more expensive or more difficult to run than others, so fewer observations may be taken.
(3) Certain treatment combinations may be of greater interest to engineers, so the engineers may want to obtain additional replications at certain design points.
(4) Some experimental data may be missing due to the physical environment.

It is of critical importance that data from experiments that have generated an unequal number of replicate points be evaluated carefully. First, the orthogonality property of main effects and interactions present in balanced data does not carry over to the unbalanced case; hence, the usual analysis of variance techniques does not apply. Consequently, the analysis of unbalanced designs is much more difficult than that for balanced designs. Second, for the balanced case, errors are independent and have constant variance. However, when it comes to the unbalanced case, the errors typically do not hold constant variance. These facts imply that the use of ordinary least squares (OLS) under the circumstances of the unbalanced case can lead to nonsensical results. To rectify these problems, we first discuss the concept of the weighted least squares method. Next we present simulation results for verification, and finally we show how this method is incorporated into robust design.

3. The method of weighted least squares

The weighted least squares (WLS) method is more efficient and useful for estimating the values of model parameters when different design points have different numbers of replicates. As suggested by the name, parameter estimation by the method of weighted least sum of squares is closely related to parameter estimation by 'regular,' 'unweighted' or 'equally-weighted' least sum of squares. This is done by attempting to give each data point the proper amount of influence over the parameter estimates, rather than giving some points more influence than they should have and giving others less. The goal is to use a procedure that treats all of the data equally.

The WLS method works by incorporating an extra nonnegative constant or weight associated with each data point into fitting models. The size of the weight reflects the accuracy of the information contained in the associated observation. The weight for each observation is given relative to the other observations and their weights. When the weighted fitting criterion is minimized to find the parameter estimates, the weights determine the contribution of each observation to the final parameter estimates. For example, if the standard deviation of the random errors in the data is not constant across all levels of the explanatory variables, using the WLS, with weights that are inversely proportional to the variance at each level of the explanatory variables, yields the most precise parameter estimates possible.

4. The proposed modeling and optimization procedures

4.1. Parameter estimation by weighted least squares

The basic assumption in the OLS regression is that $\mathrm{Var}(\epsilon_i)=\sigma^2$ for all cases in the data. This assumption is in doubt in many problems, as variances can depend on the response, on one or more of the predictors, or possibly on other factors.

In the OLS regression, we assume that the responses $\mathbf{Y}$ are normally distributed with mean $\mathbf{X}\boldsymbol{\beta}$ and variance $\sigma^2\mathbf{I}$, where $\mathbf{X}$ is a matrix of variables, $\boldsymbol{\beta}$ is a coefficient vector, and $\mathbf{I}$ is an identity matrix. That is

$$\mathbf{Y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}).$$

Note that the regression errors $\epsilon_1,\ldots,\epsilon_n$ are assumed to be i.i.d. $N(0,\sigma^2)$, where i.i.d. stands for 'independent and identically distributed'.

Now suppose that the errors have unequal variances whose proportionality constants c_i are known such that

$$\mathrm{Var}(\epsilon_i) = c_i\sigma^2, \quad i = 1, \ldots, n.$$

In matrix notation

$$\mathbf{Y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{V}),$$

where $\mathbf{V}=\mathrm{diag}[c_1,\ldots,c_n]$. Hence we have

$$\boldsymbol{\epsilon} = \mathbf{Y} - \mathbf{X}\boldsymbol{\beta} \sim N(0, \sigma^2\mathbf{V}).$$

By denoting $\mathbf{W}^{1/2}$ as $\mathrm{diag}[1/\sqrt{c_1}, \ldots, 1/\sqrt{c_n}]$, $\mathbf{W}=\mathbf{W}^{1/2}\mathbf{W}^{1/2}=\mathbf{V}^{-1}$. Then we have

$$\mathbf{W}^{1/2}\boldsymbol{\epsilon} = \mathbf{W}^{1/2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \sim N(0, \sigma^2\mathbf{I}).$$

For convenience, let us denote

$$\boldsymbol{\epsilon}^* = \mathbf{W}^{1/2}\boldsymbol{\epsilon}, \;\; \mathbf{Y}^* = \mathbf{W}^{1/2}\mathbf{Y}, \;\text{ and }\; \mathbf{X}^* = \mathbf{W}^{1/2}\mathbf{X}.$$

Thus the WLS estimator is equivalent to the OLS estimator on $\mathbf{Y}^*$ and $\mathbf{X}^*$:

$$\hat{\beta} = (\mathbf{X}^{*'}\mathbf{X}^*)^{-1}\mathbf{X}^{*'}\mathbf{Y}^* = (\mathbf{X}'\mathbf{W}^{1/2}\mathbf{W}^{1/2}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{1/2}\mathbf{W}^{1/2}\mathbf{Y} = (\mathbf{X}'\mathbf{W}\mathbf{Y})^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}.$$

4.2. Estimating process mean and variance via response surface methodology

Consider a system involving a response Y which depends on the levels of k control factors $\mathbf{x}=(x_1,x_2,\ldots,x_k)$. The following assumptions are made:

- A functional structure, $Y=g(x_1,x_2,\ldots,x_k)$, is either unknown or complicated.
- The levels of x_i for $i=1,2,\ldots,k$ are quantitative and continuous.
- The levels of x_i for $i=1,2,\ldots,k$ can be controlled by the experimenter.

Suppose that r_i replicates are taken at the ith design point. Let Y_{ij} represent the jth response at the ith design point where $i=1,2,\ldots,n$ and $j=1,2,\ldots,r_i$. The most popular estimators of the location and scale parameters are mean and variance, respectively. At the ith design point, we have the sample mean and sample variance as follows:

$$\bar{Y}_i = \frac{1}{r_i}\sum_{j=1}^{r_i} Y_{ij} \text{ and } S_i^2 = \frac{1}{r_i - 1}\sum_{j=1}^{r_i}(Y_{ij} - \bar{Y}_i)^2.$$

4.3. Incorporating the WLS into robust design

Let $\hat{m}(\mathbf{x})$ and $\hat{v}(\mathbf{x})$ represent the fitted response functions for the mean and variance of the response Y, respectively. Assuming a second-order polynomial model for the response functions, we get

$$\hat{m}(\mathbf{x}) = \hat{\beta}_0 + \sum_{i=1}^{k}\hat{\beta}_i x_i + \sum_{i=1}^{k}\sum_{j=1}^{k}\hat{\beta}_{ij}x_i x_j \text{ and } \hat{v}(\mathbf{x}) = \hat{\gamma}_0 + \sum_{i=1}^{k}\hat{\gamma}_i x_i + \sum_{i=1}^{k}\sum_{j=1}^{k}\hat{\gamma}_{ij}x_i x_j.$$

We use the sample mean and variance of Y to estimate the process mean $\hat{m}(\mathbf{x})$ and variance $\hat{v}(\mathbf{x})$, respectively.

Using the following theorem shown in most statistics textbooks, we have $\mathrm{Var}(\bar{Y}_i) \propto 1/r_i$ and $\mathrm{Var}(S_i^2) \propto 1/(r_i - 1)$. In building regression models for $m(\mathbf{x})$ and $v(\mathbf{x})$, it is more sensible to use the r_i and r_i-1 as weights, respectively. That is, $\mathbf{W}_m=\mathrm{diag}[r_1,r_2,\ldots,r_n]$ for $m(\mathbf{x})$ and $\mathbf{W}_v=\mathrm{diag}[r_1 - 1, r_2 - 1, \cdots, r_n - 1]$ for $v(\mathbf{x})$.

Theorem 1. *Let $Y_1,\ldots,Y_r$ be a random sample of size r from the probability density function $f(y)$ with a finite fourth moment and let $\mu=E(Y)$ and $\theta_k=E(Y-\mu)^k$, $k=2,3,4$. Then we have*

$$\mathrm{Var}(\bar{Y}) = \frac{1}{r}\mu \text{ and } \mathrm{Var}(S^2) = \frac{1}{r}\left(\theta_4 - \frac{r-3}{r-1}\theta_2^2\right).$$

Especially if Y_i's have independent and identical normal distribution, then $\mathrm{Var}(S^2)=2\sigma^4/(r-1)$.

The main objective of robust design is to obtain the optimum operating conditions of control factors, and this goal can be easily achieved by employing the following squared-loss optimization model

$$\text{minimize } \{\hat{m}(\mathbf{x}) - t_0\}^2 + \hat{v}(\mathbf{x}), \quad \text{subject to } x_j \in [L_j, U_j] \text{ for } j = 1, \ldots, k,$$

where t_0 is the customer-identified target value for the quality characteristic of interest, and the constraint specifies the feasible region of $\mathbf{x}$. When factorial designs with k levels are used, the constraint becomes $x_j \in [-1,1]$ for $j=1,\ldots,k$. Two items are worth mentioning. First, the following dual-response optimization model proposed by Vining and Myers (1990) can also be used for optimization purposes:

$$\text{minimize } \hat{v}(\mathbf{x}), \quad \text{subject to } \hat{m}(\mathbf{x}) = t_0 \text{ and } x_j \in [L_j, U_j] \text{ for } j = 1, \ldots, k.$$

However, the dual-response model strictly imposes a zero-bias condition, while the squared-loss model allows some bias (i.e. absolute value of the difference of $\hat{m}(x)$ and t_0). This squared-loss model often results in less variability. For detailed information regarding the squared-loss model, readers may refer to Cho et al. (2000) and Lin and Tu (1995). Second, although the quadratic fitted functions are shown above, the estimated functions can also be linear.

5. Simulation results and verification

As illustrated in Section 6, r_i responses $(Y_{i1}, \ldots, Y_{ir_i})$ based on the simulation schemes shown in Fig. 1 were generated from the distributions with $\mu(\mathbf{x}_i)$ and $\sigma(\mathbf{x}_i)$ at each control factor settings

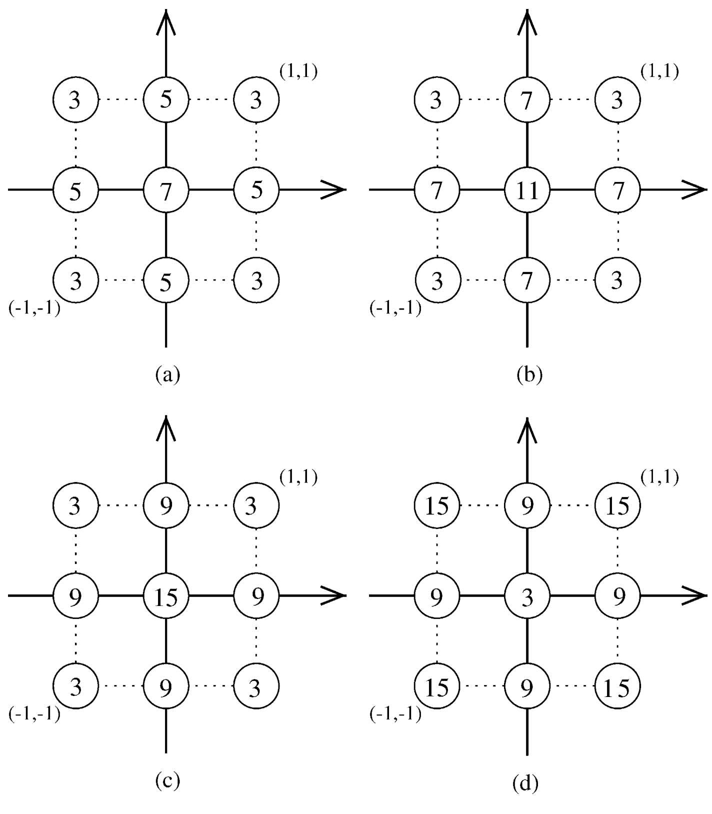


Fig. 1. Simulation schemes.

Table 1
Estimated bias, variance and MSE for each simulation scheme

Scheme	OLS			WLS			WLS/OLS		
	Bias	Var	MSE	Bias	Var	MSE	Bias	Var	MSE
(a)	3.48	6.80	18.90	3.40	6.96	18.53	0.98	1.02	0.98
(b)	3.30	5.36	16.26	3.07	5.12	14.51	0.93	0.95	0.89
(c)	3.13	5.01	14.80	2.88	4.33	12.63	0.92	0.86	0.85
(d)	2.46	6.11	12.16	2.40	5.19	10.97	0.98	0.85	0.90

$\mathbf{x}_i=(x_{i1}, x_{i2})$, $i=1,\ldots,9$. The total number of iterations is 1000, each having 9 design points, and $\mu(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ are given as follows:

$$\mu(\mathbf{x}) = 50 + 10(x_1^2 + x_2^2),$$

$$\sigma^2(\mathbf{x}) = 100 + 25(x_1^2 + x_2^2).$$

The numerical simulations are performed using the R language which is a non-commercial, open source software for statistical computing and graphics originally developed by Ihaka and Gentleman (1996). This can be obtained at no cost from: http://www.r-project.org/

Simulation was used to verify the adequacy of the WLS method when sample sizes vary. For each simulation scheme shown in Fig. 1, three statistical measures, such as bias, sample variance, and mean squared error (MSE), were considered to be decision criteria for judging the performance of OLS and WLS. Assuming that the customer-identified product target $t_0=50.0$, Table 1 shows the estimated bias, variance, and MSE of the optimal mean response $\hat{m}(\mathbf{x}^*)$.

We considered a standard 3^2 factorial design with three levels $(-1,0,+1)$. Four different schemes were tested as shown in Fig. 1 where the number in a circle represent a sample size. We then estimated process bias and variance, and obtained MSE for each simulation scheme. As shown in Table 1, the WLS approach consistently showed lower bias and variability, which resulted in lower MSE.

Table 2
Data for case study example

i	x_{i1}	x_{i2}	Y_{ir_i}							$\bar{Y}_i$	S_i^2
1	−1	−1	84.3	57.0	56.5					65.93	253.06
2	0	−1	75.7	87.1	71.8	43.8	51.6			66.00	318.28
3	1	−1	65.9	47.9	63.3					59.03	94.65
4	−1	0	51.0	60.1	69.7	84.8	74.7			68.06	170.35
5	0	0	53.1	36.2	61.8	68.6	63.4	48.6	42.5	53.46	139.89
6	1	0	46.5	65.9	51.8	48.4	64.4			55.40	83.11
7	−1	1	65.7	79.8	79.1					74.87	63.14
8	0	1	54.4	63.8	56.2	48.0	64.5			57.38	47.54
9	1	1	50.7	68.3	62.9					60.63	81.29

Table 3
Comparative summary of the mean and variance functions under the OLS and WLS

	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_{11}$	$\hat{\beta}_{22}$	$\hat{\beta}_{12}$	Bias	Var	MSE
OLS									
$\hat{m}(\mathbf{x})$	55.61	−5.63	0.32	5.04	5.00	−1.84			
$\hat{v}(\mathbf{x})$	160.65	−37.92	−79.00	−44.30	11.88	44.14			
$\mathbf{x}^*$	$(x_1^*, x_2^*) = (0.999, 0.995)$						8.46	55.56	127.14
WLS									
$\hat{m}(\mathbf{x})$	55.08	−5.76	−0.52	5.51	5.47	−1.84			
$\hat{v}(\mathbf{x})$	154.26	−39.34	−93.09	−38.31	17.87	44.14			
$\mathbf{x}^*$	$(x_1^*, x_2^*) = (0.998, 0.998)$						7.93	45.66	108.48

6. Numerical example

An injection-molding company produces bare silicon wafers as a subcontractor for a large motor corporation. The coating thickness (y) of the wafer is the most important quality characteristic. For the wafers, the target value is 50, and the key factors are mould temperature (x_1) and injection flow rate (x_2). The following 3^2 factorial design with a different number of replicates taken at each design point ($i=1,\ldots,9, j=1,\ldots,r_i$) is shown in Table 2. For the WLS, we used r_i and r_i-1 as weights to estimate $m(\mathbf{x})$ and $v(\mathbf{x})$, respectively. That is, $\mathbf{W}_m=\text{diag}[3,5,3,5,7,5,3,5,3]$ for $m(\mathbf{x})$, and $\mathbf{W}_v=\text{diag}[2,4,2,4,6,4,2,4,2]$ for $v(\mathbf{x})$.

We then obtained $\hat{m}(\mathbf{x})$ and $\hat{v}(\mathbf{x})$ as

$$\hat{m}(\mathbf{x}) = 55.08 - 5.76x_1 - 0.52x_2 + 5.51x_1^2 + 5.47x_2^2 - 1.84x_1x_2,$$

$$\hat{v}(\mathbf{x}) = 154.26 - 39.34x_1 - 93.09x_2 - 38.31x_1^2 + 17.87x_2^2 + 44.14x_1x_2.$$

By minimizing $\{\hat{m}(\mathbf{x}) - 50\}^2 + \hat{v}(\mathbf{x})$ subject to $|x_j| \leq 1$ for $j=1$ and 2, the optimum operating conditions are obtained as $(x_1^*, x_2^*) = (0.998, 0.998)$. For comparison purposes, Table 3 provides the bias, variance, MSE, and optimum operating conditions under the OLS and WLS in which the WLS shows less MSE.

7. Conclusions

The motivation for developing a new methodology was the realization that current robust design models strictly require the balanced data assumption that is not always cost-effective or practical from an engineering point of view. For example, certain treatment combinations may be more expensive or more difficult to run than others, and other combinations may be of greater interest to engineers. Under such situations, we first showed that the method of WLS is more effective than the method of OLS for maximizing the efficiency of parameter estimation. Our simulation studies showed the efficiency ratio of MSE could go up as high as 0.98. We then integrated the concept of WLS into robust design by showing mathematical modeling and optimization procedures. The central contribution of the present study is the recognition that the determination of optimum operating conditions in the robust design context must reflect recognition of the WLS for unbalanced data.

Acknowledgements

The case study was carried out in conjunction with a project grant funded by the C.S. Tech.

References

Bendell, A., Disney, J., & Pridmore, W. A. (1987). *Taguchi methods: Applications in world industry*. London: IFS Publications.

Box, G. E. P. (1985). Discussion of off-line quality control, parameter design and the Taguchi methods. *Journal of Quality Technology*, *17*, 198–206.

Cho, B. R., Kim, Y. J., Kimbler, D. L., & Phillips, M. D. (2000). An integrated joint optimization procedure for robust and tolerance design. *International Journal of Production Research*, *38*, 2309–2325.

Dehnad, K. (1989). *Quality control, robust design and the Taguchi method*. Pacific Grove, CA: Wadsworth and Brooks/Cole.

Del Castillo, E., & Montgomery, D. C. (1993). A nonlinear programming solution to the dual response problem. *Journal of Quality Technology*, *25*, 199–204.

Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, *5*, 299–314.

Kim, Y. J., & Cho, B. R. (2000). Economic considerations on parameter design. *Quality and Reliability Engineering International*, *16*, 501–504.

Kim, Y. J., & Cho, B. R. (2002). Development of priority-based robust design. *Quality Engineering*, *14*(3), 355–363.

Lin, D. K. J., & Tu, W. (1995). Dual response surface optimization. *Journal of Quality Technology*, *27*, 34–39.

Myers, R. H., Khuri, A. I., & Vining, G. G. (1992). Response surface alternatives to the Taguchi robust design problem. *American Statistician*, *46*, 131–139.

Myers, R. H., & Montgomery, D. C. (2002). *Response surface methodology* (2nd ed.). New York: Wiley.

Pignatiello, J., & Ramberg, J. S. (1991). Top ten triumphs and tragedies of Genichi Taguchi. *Quality Engineering*, *4*, 221–225.

Tiao, G. C., Bisgaard, S., Hill, W. J., Peña, D., & Stigler, S. M. (Eds.). (2000). *Box on quality and discovery with design, control and robustness*. New York: Wiley.

Vining, G. G., & Myers, R. H. (1990). Combining Taguchi and response surface philosophies: A dual response approach. *Journal of Quality Technology*, *22*, 38–45.