Chapter 1

Least squares

29 August 2016

1.1 Minimimize sums of squares

The least squares method was first proposed in the early years of the 19th century (or maybe earlier—Stigler, 1986, Chapter 1). Statistician often think of it as just a way to fit a statistical model, even though it makes sense without any mention of randomness or a model.

As a problem in computation, least squares starts from $y = [y_1, \ldots, y_n]$, a column vector of numbers (an element of $\mathbb{R}^n$), and some finite set of "predictors", column vectors $x_j = [x_{1j}, \ldots, x_{nj}]$ for $j = 1, \ldots, p$. We seek numbers $b_1, \ldots, b_p$ to minimize the sum of squared differences:

$$\sum\nolimits_{i \le n} \Big(y_i - \sum\nolimits_{j \le p} x_{ij} b_j \Big)^2 .$$

If we write $\|z\| = \sqrt{z_1^2 + \cdots + z_n^2}$ for the length of a vector $z = [z_1, \ldots, z_n]$ in $\mathbb{R}^n$, then the task can be reexpressed as:

$$\text{minimize } \|y - x_1 b_1 - \cdots - x_p b_p\|^2 \text{ with respect to } b. \tag{1.1}$$

That is, we seek a linear combination of the vectors $x_1, \ldots, x_p$ that best approximates y, in the sense of minimizing the usual Euclidean distance. More compactly, we can take the x_j's as the columns of an $n \times p$ matrix and think of $b = [b_1, \ldots, b_p]$ as a vector in $\mathbb{R}^p$, then minimize $\|y - Xb\|^2$. We could also think of $y_1, \ldots, y_n$ as measurements on each of n individuals and regard the ith row, w_i', of X as providing auxiliary information about individual i, then minimize $\sum_{i \le n} (y_i - w_i' b)^2$.

Remark. If the last paragraph gave you a headache then you might want to look at the first two chapters of the Axler (2015) book. It is pretty painful trying to learn about linear models if one does not know any linear algebra.

It is not hard to find a set of linear equations for b whose solution achieves the minimum. The solution is not unique unless the vectors $x_1, \ldots, x_p$ are linearly independent, that is, if none of them can be written as a linear combination of the others (cf. Axler, 2015, Section 2A). In some simple cases it is possible to write out, without recourse to matrices, a closed form expression for the solution. It used to be a popular source of pleasure for statistical sadists to make students memorize that expression, despite the fact that computers are much better than humans at memorizing and applying formulas.

Using **R** it is particularly easy to find least squares solutions. Here is a fake example:

```
set.seed(0)   # gardening time
xx <- matrix(1:20,ncol=2) # Create a 10 by 2 matrix
xx[1:3,] # first three rows of xx
##      [,1] [,2]
## [1,]    1   11
## [2,]    2   12
## [3,]    3   13
# The columns of xx are known as xx[,1] and xx[,2] to R.
yy <- rnorm(10) # just noise
```

Now solve.

```
out1 <- lm(yy ~ -1 + xx) # solve the least squares problem,
      # putting lots of info into a new object
names(out1)  # lots of good stuff in here
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
# For example,
out1$coefficients
##         xx1         xx2
## -0.07498644  0.04976448
# too many decimal places
```

In this case the minimizing value of b is $\widehat{b} = [-0.07, 0.05]$.

Remark. The -1 in the $lm()$ stops **R** from trying to be helpful by adding a vector of 1's as a third column to X, even though it is often a good idea. (It adds an intercept term to a regression.) If you want to see the effect, try:

```
# summary(lm(yy ~ -1 + xx))
# summary(lm(yy ~ xx))
```

without the comment characters (#) at the start of the lines. You will see that **R** has included an intercept term, by adding a column of 1's to the X matrix, in the second case. Don't worry about the details. You'll learn in a week or so what it all means.

Usually it is better to keep all the data together in one object, a "data frame":

```
mydata <- data.frame(y=yy,xx)
mydata[1:3,]
##            y X1 X2
## 1  1.2629543  1 11
## 2 -0.3262334  2 12
## 3  1.3297993  3 13
out2 <- lm(y ~ -1 + X1 + X2, data=mydata)
# out2 <- lm(y ~ -1 + . , data=mydata) does the same thing
round(out2$coeff,4)
##      X1      X2
## -0.0750  0.0498
```

1.2 Geometry

The vectors $x_1, \ldots, x_p$ span some subspace $\mathcal{X}$ of $\mathbb{R}^n$. (That is, the set of all vectors of the form $x_1 b_1 + \ldots x_p b_p$ is a vector subspace of $\mathbb{R}_n$). The least squares method minimizes $\|y - x\|$ as x runs through all vectors in $\mathcal{X}$. The minimimizing vector in $\mathcal{X}$ is often denoted by $\widehat{y}$. It is sometimes called the ***fitted vector***. As you will see, it is easy to find $\widehat{y}$ if we work in a good coordinate system. We then have the task of writing $\widehat{y}$ as a linear combination $\widehat{b}_1 x_1 + \cdots + \widehat{b}_p x_p$, which also turns out to be easy if we choose a good coordinate system.

If the x_j's are linearly independent (Axler, 2015, Section 2.17) then the subspace $\mathcal{X}$ has ***dimension*** p (and X has ***rank*** p; the matrix is of ***full rank***). If at least one of the x_j's is a linear combination of some of the others then $\mathcal{X}$ has a dimension smaller than p (and X is not of full rank).

> **Remark.** There are many linear models for which it makes sense to choose linearly dependent x_j's. Stay tuned for ANOVA.

Axler (2015, Chapter 6) would come in handy if the next paragraph intimidates you.

Remember that $\mathbb{R}^n$ can be equipped with an inner product (also known as a dot product). For column vectors $u = [u_1, \ldots, u_n]$ and $v = [v_1, \ldots, v_n]$,

$$\langle u, v \rangle = u \cdot v = u'v = \sum\nolimits_{i \leq n} u_i v_i.$$

In particular, $\|u\| = \sqrt{\langle u, u \rangle}$. A vector u is said to be a unit vector if $\|u\| = 1$. Two vectors, u and v, are said to be orthogonal if $\langle u, v \rangle = 0$. A set of vectors $\{q_1, \ldots, q_m\}$ is said to be an ***orthonormal basis*** for a subspace $\mathcal{X}$ if:

(i) Each q_i is a unit vector and $\langle q_i, q_j \rangle = 0$ for all $i \neq j$.

(ii) Each vector x in can be written as a linear combination of $q_1, \ldots, q_m$.

The second property means that, for each x in $\mathcal{X}$ there exist numbers $t_1, \ldots, t_m$ such that

$$x = q_1 t_1 + \cdots + q_m t_m$$

The coefficients t_j are determined by

$$\langle x, q_j \rangle = \langle q_1 t_1 + \cdots + q_m t_m, q_j \rangle = t_1 \langle q_1, q_j \rangle + \cdots + t_j \langle q_j, q_j \rangle + \cdots + t_n \langle q_n, q_j \rangle.$$

In the last sum, $\langle q_j, q_j\rangle = 1$ (because q_j is a unit vector) and the other inner products are zero (by orthogonality). Thus $t_j = \langle x, q_j\rangle$.

The key fact is:

For each subspace $\mathcal{X}$ of $\mathbb{R}^n$ of dimension m there exists an orthonormal basis $\{q_j : 1 \le j \le n\}$ (not unique) for $\mathbb{R}^n$, such that $q_1, \ldots, q_m$ is an orthonormal basis for $\mathcal{X}$. The remaining vectors, $q_{m+1}, \ldots, q_n$ provide an orthonormal basis for the subspace

$$\mathcal{X}^\perp = \{z \in \mathbb{R}^n : \langle x, z\rangle = 0 \text{ for all } x \text{ in } \mathcal{X}\},$$

the orthogonal complement of $\mathcal{X}$.

It is a trivial task to find $\widehat{y}$ if y and x are expressed in this orthonormal basis.

$$\begin{aligned} y &= s_1 q_1 + \ldots s_m q_m + (s_{m+1} q_{m+1} + \ldots s_n q_n) \\ x &= t_1 q_1 + \ldots t_m q_m \\ \|y - x\|^2 &= (s_1 - t_1)^2 + \cdots + (s_1 - t_1)^2 + \left(s_{m+1}^2 + \cdots + s_n^2\right) \end{aligned} \tag{1.2}$$

Clearly the solution is given by $t_i = s_i$ for $1 \le i \le m$. The ***fitted vector*** $\widehat{y}$ equals $s_1 q_1 + \ldots s_m q_m$. The ***residual vector*** is defined as $r = y - \widehat{y} = s_{m+1} q_{m+1} + \ldots s_n q_n$, an element of $\mathcal{X}^\perp$. The fitted vector and the residual vector are orthogonal. The decomposition $y = \widehat{y} + r$ is the unique representation of y as a sum of a vector in $\mathcal{X}$ and a vector in $\mathcal{X}^\perp$. The fitted vector is also called the ***orthogonal projection*** of y onto $\mathcal{X}$ and the residual is called the projection of y orthogonal to $\mathcal{X}$ (or the orthogonal projection of y onto $\mathcal{X}^\perp$).

<1.3> **Example.** Most introductory courses discuss the problem of fitting a straight line by least squares: We have vectors y and $z = [z_1, \ldots, z_n]$ in $\mathbb{R}^n$; we seek constants c and d to minimize $\sum_{i \le n}(y_i - c - d z_i)^2$.

The problem fits into the framework of (1.1) with $x_1 = \mathbb{1}$, a column of 1's, and $x_2 = z$—that is, $X = (\mathbb{1}, z)$—and b equal to the column vector $[c, d]$.

For the q_i's choose q_1 as a unit vector in the direction of x_1 and q_2 as a unit vector in the direction of $x_2 - \langle x_2, q_1\rangle q_1$, which is orthogonal to q_1. Note

that $\langle x_2, q_1\rangle = n^{-1/2}\sum_i z_i = \overline{z}\sqrt{n}$, so that

$$q_1 = n^{-1/2}\mathbb{1}$$
$$q_2 = \frac{x_2 - \langle x_2, q_1\rangle q_1}{\|x_2 - \langle x_2, q_1\rangle q_1\|} = \frac{z - \overline{z}\mathbb{1}}{\ell} \qquad \text{where } \ell := \sqrt{\sum\nolimits_i (z_i - \overline{z})^2}\,.$$

We do not need to determine $q_3, \ldots, q_n$ explicitly in order to determine the least squares fit. Indeed, with

$$s_1 = \langle y, q_1\rangle = n^{1/2}\overline{y} \quad \text{AND} \quad s_2 = \langle y, q_2\rangle = \sum\nolimits_i y_i(z_i - \overline{z})/\ell$$

we have

$$\widehat{y} = s_1 q_1 + s_2 q_2 = \widehat{c}\mathbb{1} + \widehat{d}z$$
$$\text{where } \widehat{d} = \frac{\sum_i y_i(z_i - \overline{z})}{\sum_i (z_i - \overline{z})^2} \quad \text{AND} \quad \widehat{c} = \overline{y} - \widehat{d}\overline{z}.$$

Remark. Obviously something goes wrong if $\sum_i (z_i - \overline{z})^2 = 0$. That happens only when z is a constant multiple of $\mathbb{1}$, which means that z and $\mathbb{1}$ are linearly dependent and the coefficients c and d are not uniquely determined. Put another way, the rank of the matrix X and the dimension of the subspace $\mathcal{X}$ both equal 1. For uniquely determined c and d we need the rank and the dimension equal to 2.

In matrix notation, provided z is not a constant multiple of $\mathbb{1}$,

$$Q = (q_1, q_2) = (\mathbb{1}, z)W = XW \tag{1.4}$$
$$\text{where } W = \begin{pmatrix} w_{11} & w_{12} \\ 0 & w_{22} \end{pmatrix} := \begin{pmatrix} n^{-1/2} & -\overline{z}/\ell \\ 0 & 1/\ell \end{pmatrix}.$$

The zero in the south-west corner makes it easy to find the inverse matrix $R = W^{-1}$. We want

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = WR = \begin{pmatrix} w_{11}r_{11} + w_{12}r_{21} & w_{11}r_{12} + w_{12}r_{22} \\ w_{22}r_{21} & w_{22}r_{22} \end{pmatrix}.$$

with solution

$$r_{22} = 1/w_{22} = \ell$$
$$r_{21} = 0/w_{22} = 0$$
$$r_{11} = (1 - 0)/w_{11} = n^{1/2}$$
$$r_{12} = -w_{12}r_{22}/w_{11} = n^{1/2}\overline{z}.$$

That is

$$R = \begin{pmatrix} n^{1/2} & n^{1/2}\overline{z} \\ 0 & \ell \end{pmatrix}$$

and $X = XWW^{-1} = QR$.

Remark. If I had started with a much larger ***upper triangular matrix***, a square matrix with zeros everywhere below the diagonal, you would have seen clearly the pattern that gives the method the name ***back-substitution*** (Golub and Van Loan, 2013, Section 3.1). **R** knows how to back-substitute; you do not have to go through the calculations.

The least squares calculations can be rewritten as

$$QR\widehat{b} = X\widehat{b} = \widehat{y} = \langle y, q_1\rangle q_1 + \langle y, q_2\rangle q_2 = QQ'y,$$

so that $\widehat{b} = R^{-1}Q'y$. (Orthonormality gives $Q'Q = I_2$.) The $n \times n$ matrix $H = QQ'$ is often called the ***hat matrix***. (Where do you think that name came from?) It is the matrix for orthogonal projection onto the $\mathcal{X}$ subspace. Also

$$\widehat{b} = R^{-1}Q'y = (R'R)^{-1}(QR)'y = (X'X)^{-1}X'y.$$

The last equality gives the traditional textbook solution to the "normal equations", $X'(y - X\widehat{b}) = r'\widehat{y} = 0$. Unfortunately this solution makes no sense when X is not of full rank. (The matrix R would be singular, that is neither R nor $X'X$ has an inverse.)

□

1.3 QR decompositions and least squares

The method described in Example 1.3 works more generally. For $n \geq p$, every $n \times p$ matrix $X = (x_1, \ldots, x_p)$ can be written in the form $X = QR$, where $Q = (q_1, \ldots, q_p)$ is an $n \times p$ matrix whose columns are orthogonal unit vectors and R is a $p \times p$ upper triangular matrix (the elements below the diagonal are all zero):

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \ldots & r_{1p} \\ 0 & r_{22} & r_{23} & \ldots & r_{2p} \\ 0 & 0 & r_{33} & \ldots & r_{3p} \\ \vdots & & & & \\ 0 & 0 & 0 & \ldots & r_{pp} \end{pmatrix}$$

That is,

$$x_1 = q_1 r_{11}$$
$$x_2 = q_1 r_{12} + q_2 r_{22}$$
$$\vdots \tag{1.5}$$
$$x_p = q_1 r_{1p} + q_2 r_{2p} + \cdots + q_p r_{pp} \tag{1.6}$$

Such a representation could be constructed by an analog of the procedure in Example 1.3 (the Gram-Schmidt procedure) or, as **R** does, by an elegant method using Householder reflections (Dongarra et al., 1993, page 9.13).

The diagonal elements r_{jj} for $1 \leq j \leq p$ are all nonzero if and only if the matrix X has full rank. In that case, the q_j's can all be expressed (uniquely) as linear combinations of the x_j's and $\{q_1, \ldots, q_p\}$ is an orthonormal basis for $\mathcal{X}$.

When X is not of full rank, **R** can still make a least squares fit, but you need to be careful in interpreting the output. Here is an example where I ensure complications by deliberately making the third column of X a linear combination of two other columns, then **R** makes things even worse (so to speak) by adding in a column of ones.

```
mydata3 <- data.frame(y=yy,
        x1=1:10,x2= 11:20, x3= 0.5*(1:10)-3*(11:20))
out3 <- lm(y ~ ., data=mydata3)
summary(out3)
```

```
##
## Call:
## lm(formula = y ~ ., data = mydata3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8863 -0.7277 -0.1167  0.8544  2.1592
##
## Coefficients: (2 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.49764    0.87159   0.571    0.584
## x1          -0.02522    0.14047  -0.180    0.862
## x2                NA         NA      NA       NA
## x3                NA         NA      NA       NA
```

```
##
## Residual standard error: 1.276 on 8 degrees of freedom
## Multiple R-squared:  0.004014,Adjusted R-squared:  -0.1205
## F-statistic: 0.03224 on 1 and 8 DF,  p-value: 0.862
```

What happened? **R** figured out that (within some level of numerical accuracy): x_2 and x_3 are linear combinations of $\mathbb{1}$ and x_1; the matrix $X = (\mathbb{1}, x_1, x_2, x_3)$ has rank 2; the subspace $\mathcal{X}$ has dimension 2, not 4; **R** only needs two orthogonal unit vectors to span $\mathcal{X}$; but it still gave you four orthogonal unit vectors because it thought that's what you wanted.

```
out3$rank
## [1] 2
round(qr.R(out3$qr),2) # notice the zeros on the diagonal
##   (Intercept)     x1     x2     x3
## 1       -3.16 -17.39 -49.02 138.35
## 2        0.00   9.08   9.08 -22.71
## 3        0.00   0.00   0.00   0.00
## 4        0.00   0.00   0.00   0.00
Q1 <- qr.Q(out3$qr)
round(Q1,2)[1:2,]
##       [,1]  [,2]  [,3]  [,4]
## [1,] -0.32 -0.50 -0.34 -0.32
## [2,] -0.32 -0.39 -0.24 -0.12
Q2 <- Q1[,1:2] # just the first two columns, 10 by 2
HAT <- Q2 %*% t(Q2) # a 10 by 10 matrix
round(HAT[1:2,],2)
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]  [,8]  [,9] [,10]
## [1,] 0.35 0.29 0.24 0.18 0.13 0.07 0.02 -0.04 -0.09 -0.15
## [2,] 0.29 0.25 0.21 0.16 0.12 0.08 0.04 -0.01 -0.05 -0.09
thefit <- HAT %*% mydata3$y # a 10 by 1 matrix
round( rbind(t(HAT %*% mydata3$y),out3$fit) ,3)
##          1     2     3     4     5     6     7     8     9    10
## [1,] 0.472 0.447 0.422 0.397 0.372 0.346 0.321 0.296 0.271 0.245
## [2,] 0.472 0.447 0.422 0.397 0.372 0.346 0.321 0.296 0.271 0.245
```

The hat matrix $H = Q_2 Q_2'$ is symmetric and $HH = Q_2 Q_2' Q_2 Q_2' = H$. It projects vectors in $\mathbb{R}^{10}$ onto the two-dimensional subspace $\mathcal{X}$ spanned by the columns of X or by the first two columns of the Q matrix. In fact, **R** would

be just as happy to give a complete orthonormal basis for $\mathbb{R}^{10}$:

```
Q3 <- qr.Q(out3$qr,complete=T)
round(Q1,2)[1:3,]
##       [,1]  [,2]  [,3]  [,4]
## [1,] -0.32 -0.50 -0.34 -0.32
## [2,] -0.32 -0.39 -0.24 -0.12
## [3,] -0.32 -0.28  0.90 -0.08
round(Q3,2)[1:3,]
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
## [1,] -0.32 -0.50 -0.34 -0.32 -0.31 -0.29 -0.28 -0.26 -0.24 -0.23
## [2,] -0.32 -0.39 -0.24 -0.12 -0.01  0.10  0.22  0.33  0.45  0.56
## [3,] -0.32 -0.28  0.90 -0.08 -0.07 -0.05 -0.03 -0.01  0.00  0.02
```

References

Axler, S. J. (2015). *Linear Algebra Done Right* (Third ed.). Undergraduate Texts in Mathematics. Springer.

Dongarra, J. J., C. B. Moler, J. R. Bunch, and G. W. Stewart (1993). *Linpack Users' Guide*. Society for Industrial and Applied Mathematics.

Golub, G. H. and C. F. Van Loan (2013). *Matrix Computations* (4th ed.). Johns Hopkins University Press.

Stigler, S. M. (1986). *The History of Statistics: The Measurement of Uncertainty Before 1900*. Cambridge, Massachusetts: Harvard University Press.