Regression - I

Prof. Dan A. Simovici

UMB
1 Brief Remainder of Linear Algebra

2 Linear Regression

3 Examples in R
Most of this preliminary discussion is centered around the notion of matrix rank.
Let $A \in \mathbb{R}^{m \times n}$ be a matrix. The null space of $A$ is the subspace of $\mathbb{R}^n$ defined by

$$\text{Nullsp}(A) = \{ x \in \mathbb{R}^n \mid Ax = 0_m \}.$$ 

The range of $A$ is the subspace of $\mathbb{R}^m$ defined as

$$\text{Ran}(A) = \{ y \in \mathbb{R}^m \mid y = Ax \}.$$ 

The rank of $A$ is the number $\text{rank}(A)$ that is dimension of $\text{Ran}(A)$, that is, the size of the largest linearly independent set in $\text{Ran}(A)$. 
If $A \in \mathbb{R}^{m \times n}$, the transposed matrix is $A' \in \mathbb{R}^{n \times m}$.

The *inner product* of two vectors $x, y$ in $\mathbb{R}^p$ is the number $(x, y) = x'y$. 
Theorem

Let $A \in \mathbb{R}^{m \times n}$, and $x \in \mathbb{R}^n$, and $y \in \mathbb{R}^m$. We have $(Ax, y) = (x, A'y)$.

Proof: We have $(Ax, y) = (Ax)'y = x'A'y$ and $(x, A'y) = x'(A'y)$ and these numbers are equal by the associativity.
If \( a \in \mathbb{R}^{m \times n} \) and \( A = BC \), where \( B \in \mathbb{R}^{m \times r} \) and \( C \in \mathbb{R}^{r \times n} \), then

- the \( i^{th} \) row of \( A \) is a linear combination of the \( r \) rows of \( C \) with coefficients from the \( i^{th} \) row of \( B \);
- the \( j^{th} \) column of \( A \) is a linear combination of the \( r \) columns of \( B \) with coefficients from the \( j^{th} \) row of \( C \);
If any collection of rows $\vec{c}_1, \ldots, \vec{c}_r$ spans the row space of $A$ and an $r \times n$ matrix $C$ can be formed by taking these vectors as its rows; then, the $i^{th}$ row of $A$ is a linear combination of the rows of $C$, say $\vec{a}_i = b_{i1}\vec{c}_1 + \cdots + b_{ir}\vec{c}_r$. This means that $A = BC$, where $B = (b_{ij})$ is the $m \times r$ matrix, where the $i^{th}$ row is $\vec{b}_i = (b_{i1}, \ldots, b_{ir})$;

similarly, if any $r$ column vectors span the column space of $A$ and $B$ is the $m \times r$ matrix formed by these columns, then the $r \times n$ matrix $C$ formed from appropriate coefficients satisfies $A = BC$. 
Theorem

If $A \in \mathbb{R}^{m \times n}$, then the row rank of $A$ is equal to the column rank of $A$.

Proof: If $A = O_{m \times n}$, then the row rank and the column rank are 0; otherwise, let $r$ be the smallest positive integer such that there exists $B \in \mathbb{R}^{m \times r}$ and $C \in \mathbb{R}^{r \times n}$ such that $A = BC$. Since the $r$ rows of $C$ form a minimal spanning set of the row space of $A$ and the $r$ columns of $B$ form a minimal spanning set of the column space of $A$, row and column ranks are both $r$. 
**Theorem**

Let \( A \in \mathbb{R}^{m \times n} \). We have

\[
\dim(\text{Nullsp}(A)) + \dim(\text{Ran}(A)) = n.
\]

Suppose that \( \{e_1, \ldots, e_m\} \) is a basis for \( \text{Nullsp}(A) \subseteq \mathbb{R}^n \). Extend this base to a base for \( \mathbb{R}^n \): \( \{e_1, \ldots, e_m, e_{m+1}, \ldots, e_n\} \). Any \( v \in \mathbb{R}^n \) can be written as

\[
v = v_1 e_1 + \cdots + v_m e_m + v_{m+1} e_{m+1} + \cdots + v_n e_n,
\]

hence

\[
Av = v_{m+1} Ae_{m+1} + \cdots + v_n Ae_n.
\]

Therefore, \( \{Ae_{m+1}, \ldots, Ae_n\} \) spans \( \text{Ran}(A) \). This set is linearly independent, so it is a base for \( \text{Ran}(A) \) and thus, \( \dim(\text{Ran}(A)) = n - m \).
Definition

A matrix $A$ is invertible if there exists a matrix $A^{-1}$ such that $AA^{-1} = A^{-1}A = I_n$.

Theorem

If $A \in \mathbb{R}^{n \times n}$ is invertible, then $\text{rank}(A) = n$. 
$B \in \mathbb{R}^{m \times n}$ is a **full-rank matrix** if $\text{rank}(B) = \min\{m, n\}$.

Let $B \in \mathbb{R}^{m \times n}$ be a full-rank matrix such that $m > n$, so $\text{rank}(B) = n$. The symmetric square matrix $B'B \in \mathbb{R}^{n \times n}$ has the same rank $n$ as the matrix $B$ because $\text{Nullsp}(B'B) = \text{Nullsp}(B)$. This makes $B'B$ an invertible matrix, that is, there exists $(B'B)^{-1}$. 
Experimental Setting

The *design matrix* of an experiment is put together as follows:

- The results of a series of $m$ experiments are the components of a vector $\mathbf{y} \in \mathbb{R}^m$.
- For the $i^{th}$ experiment, the values $b_{i1}, \ldots, b_{in}$ of the input variables $x_1, \ldots, x_n$ are placed in the $i^{th}$ row of a matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$.
- The outcome of the $i^{th}$ experiment $y_i$ is supposed to be a linear function of the values $b_{i1}, \ldots, b_{in}$ of $x_1, \ldots, x_n$, that is

$$y_i = b_{i1} r_1 + \cdots + b_{in} r_n.$$
The variables $x_1, \ldots, x_n$ are referred to as the *regressors*. The values assumed by the variable $x_j$ in the series of $m$ experiments, $b_{1j}, \ldots, b_{mj}$ have been placed in the $j^{\text{th}}$ column $b_j$ of the matrix $B$.

\[
\begin{array}{cccccc}
1 & x_1 & \cdots & x_j & \cdots & x_n & y \\
 1 & b_{11} & \cdots & b_{1j} & \cdots & b_{1n} & y_1 \\
 \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots \\
 m & b_{m1} & \cdots & b_{mj} & \cdots & b_{mn} & y_m \\
\end{array}
\]
Linear regression assumes the existence of a linear relationship between the outcome of an experiment and values of variables that are measured during the experiment.

In general there are more experiments than variables, that is, we have $n < m$. In matrix form we have $\mathbf{y} = \mathbf{B}\mathbf{r}$, where $\mathbf{B} \in \mathbb{R}^{m \times n}$ and $\mathbf{r} \in \mathbb{R}^n$. The problem is to determine $\mathbf{r}$, when $\mathbf{B}$ and $\mathbf{y}$ are known. Since $n < m$, this linear system is inconsistent, but is is possible to obtain an approximative solution by determining $\mathbf{r}$ such that $\| \mathbf{y} - \mathbf{B}\mathbf{r} \|$ is minimal. This amounts to approximating $\mathbf{y}$ by a vector in the subspace $\text{Ran}(\mathbf{B})$ generated by the columns of the matrix $\mathbf{B}$. 
The columns $b_1, \ldots, b_n$ of the matrix $B$ are referred to as the *regressors*; the linear combination $r_1 b_1 + \cdots + r_n b_n$ is the *regression of $y$ onto the regressors* $b_1, \ldots, b_n$.

A variant of the previous model is to assume that $y$ is affinely dependent on $b_1, \ldots, b_q$, that is,

$$y = r_0 + r_1 b_1 + \cdots + r_n b_n,$$

and we seek to determine the coefficients $r_0, r_1, \ldots, r_n$. The term $r_0$ is the *bias* of the model. The dependency of $y$ on $b_1, \ldots, b_n$ can be homogenized by introducing a dummy vector $b_0$ having all components equal to 1, which gives

$$y = r_0 b_0 + r_1 b_1 + \cdots + r_n b_n,$$

as the defining assumption of the model.
As we stated before, if the linear system $B r = y$ has no solution $r$, the “next best thing” is to find a vector $r \in \mathbb{R}^n$ such that

$$\| B r - y \|_2 \leq \| B w - y \|_2$$

for every $w \in \mathbb{R}^n$. This approach is known as the least square method. We will refer to the triple $(B, r, y)$ as an instance of the least square problem.
• Note that $Br \in \text{range}(B)$ for any $r \in \mathbb{R}^n$. Thus, solving this problem amounts to finding a vector $Br$ in the subspace $\text{range}(B)$ such that $Br$ is as close to $y$ as possible.

• Let $B \in \mathbb{R}^{m \times n}$ be a full-rank matrix such that $m > n$, so $\text{rank}(B) = n$. The symmetric square matrix $B'B \in \mathbb{R}^{n \times n}$ has the same rank $n$ as the matrix $B$. Therefore, the system $(B'B)r = B'y$ a unique solution $r = (B'B)^{-1}B'y$. Moreover, $B'B$ is positive definite because $r'B'Br = (Br)'Br = \|Br\|_2^2 > 0$ for $r \neq 0_n$. 
Theorem

Let $B \in \mathbb{R}^{m \times n}$ be a full-rank matrix such that $m > n$ and let $y \in \mathbb{R}^m$. The unique solution $r = (B' B)^{-1} B' y$ of the system $(B' B)r = B' y$ equals the projection of the vector $y$ on the subspace Ran($B$).
Proof

The $n$ columns of the matrix $B = (b_1 \cdots b_n)$ constitute a basis of the subspace $\text{range}(B)$. Therefore, we seek the projection $c$ of $y$ on $\text{range}(B)$ as a linear combination of the columns of $B$, $c = Bt$, which allows us to reduce this problem to a minimization of the function

$$f(t) = \| Bt - y \|_2^2$$

$$= (Bt - y)'(Bt - y) = (t'B' - y')(Bt - y)$$

$$= t'B'Bt - y'Bt - t'B'y + y'y.$$  

The necessary condition for the minimum is

$$(\nabla f)(t) = 2B'Bt - 2B'y = 0,$$  

which implies $B'Bt = B'y$.  


The linear system \((B' B)t = B'y\) is known as the system of normal equations of \(B\) and \(y\).
Suppose now that $B \in \mathbb{R}^{m \times n}$ has rank $k$, where $k < \min\{m, n\}$, and $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ are orthonormal matrices such that $B$ can be factored as $B = UMV'$, where

$$
M = \begin{pmatrix}
R & O_{k, n-k} \\
O_{m-k, k} & O_{m-k, n-k}
\end{pmatrix} \in \mathbb{R}^{m \times n},
$$

$R \in \mathbb{R}^{k \times k}$, and $\text{rank}(R) = k$.

For $y \in \mathbb{R}^m$ define $c = U'y \in \mathbb{R}^m$ and let $c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$, where $c_1 \in \mathbb{R}^k$ and $c_2 \in \mathbb{R}^{m-k}$. Since $\text{rank}(R) = k$, the linear system $Rz = c_1$ has a unique solution $z_1$. 
Theorem

All vectors \( r \) that minimize \( \| Br - y \|_2 \) have the form

\[
r = V \begin{pmatrix} z \\ w \end{pmatrix},
\]

for an arbitrary \( w \).
Proof

We have

\[ \| Br - y \|_2^2 = \| UMV'r - UU'y \|_2^2 \]
\[ = \| U(MV'r - U'y) \|_2^2 = \| MV'r - U'y \|_2^2 \]
\[ \text{(because multiplication by an orthonormal matrix} \]
\[ \text{is norm-preserving)} \]
\[ = \| MV'r - c \|_2^2 = \| My - c \|_2^2 \]
\[ = \| Rz - c_1 \|_2^2 + \| c_2 \|_2^2, \]

where \( z \) consists of the first \( r \) components of \( y \). This shows that the minimal value of \( \| Br - y \|_2^2 \) is achieved by the solution of the system \( Rz = c_1 \) and is equal to \( \| c_2 \|_2^2 \). Therefore, the vectors \( r \) that minimize \( \| Br - y \|_2^2 \) have the form \( \begin{pmatrix} z \\ w \end{pmatrix} \) for an arbitrary \( w \in \mathbb{R}^{n-r} \).
Instead of the Euclidean norm we can use the $\| \cdot \|_\infty$. Note that we have $t = \| Br - y \|_\infty$ if and only if $-t1 \leq Br - y \leq t1$, so finding $r$ that minimizes $\| \cdot \|_\infty$ amounts to solving a linear programming problem: minimize $t$ subjected to the restrictions $-t1 \leq Br - y \leq t1$. 
An optimization approach to linear regression seeks $\mathbf{r} \in \mathbb{R}^n$ that minimizes the square loss function $L : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ defined as

$$L(\mathbf{r}) = \frac{1}{n} \sum_{j=1}^{n} ((\mathbf{b}_j, \mathbf{r}) - y_j)^2.$$ 

Since

$$\frac{\partial L}{\partial r_k} = \frac{2}{n} \sum_{j=1}^{n} ((\mathbf{b}_j, \mathbf{r}) - y_j)b_k,$$

it follows that the gradient of $L$ is

$$(\nabla L)(\mathbf{r}) = \frac{2}{n} \sum_{j=1}^{n} ((\mathbf{b}_j, \mathbf{r}) - y_j)\mathbf{b}_j.$$ 

The condition $(\nabla L)(\mathbf{r}) = 0$ that is necessary for the optimum amounts now to $(B'B)\mathbf{r} = B'y$, that is to the system of normal equations of $B$ and $y$. 

Consider the simple data cars set that comes with the basic distribution of R.

> data(cars)

> str(cars)

will produce a brief description of cars that looks like:

> str(cars)

'data.frame': 50 obs. of 2 variables:
  $ speed: num 4 4 7 7 8 9 10 10 10 11 ...
  $ dist : num 2 10 4 22 16 10 18 26 34 17 ...

To produce a scatter plot of the data we write:

```r
> pdf("scatter.pdf")

> str(cars)
'data.frame': 50 obs. of 2 variables:
$ speed: num 4 4 7 7 8 9 10 10 10 11 ...
$ dist : num 2 10 4 22 16 10 18 26 34 17 ...

> pdf.off()
```

This results in the plot shown on the next slide.
This results in the scatter plot graph:
LOESS (Locally Weighted Scatterplot Smoothing), is a popular tool used in regression analysis that creates a smooth line through a timeplot or scatter plot to help you to see relationship between variables and foresee trends.
To build the linear models we need to use the function `lm()` which takes two arguments:

- a formula, and
- data.

The data is typically a `data.frame` object and the formula is a object of class `formula`. 
The function `lm` generates the *regression model*. Its arguments are a *formula* and a *data set*. For the next call of `lm` the formula is `dist ~ speed` indicating that we seek a dependency of `dist` on `speed`:

```r
# build linear regression model on full data
> linearMod <- lm(dist ~ speed, data=cars)
```
print(linearMod)

Call:

lm(formula = dist ~ speed, data = cars)

Coefficients:

(Intercept) speed

-17.579 3.932

We have established the relationship between the predictor speed and response dist:

\[ \text{dist} = 17.579 + 3.932 \times \text{speed} \]
The data set and the regression line are drawn next using the R code:

```r
> pdf("cars.pdf")
> plot(cars)
> abline(-17.579,3.932,col="red")
> dev.off()
```
This results in the drawing:
This enables us to make predictions. For example if speed is 100, the dist will be

\[ 17.579 + 3.932 \times 100 = 50.899 \]
Multiple Linear Regression

Example

The data set mtcars is part of the basic R:

```r
> data(mtcars)
> str(mtcars)
'data.frame': 32 obs. of 11 variables:
$ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ... 6 6 4 6 8 6 8 4 4 6 ...
$ disp: num 160 160 108 258 360 ...
$ hp : num 110 110 93 110 175 105 245 62 95 123 ...
$ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
$ wt : num 2.62 2.88 2.32 3.21 3.44 ...
$ qsec: num 16.5 17 18.6 19.4 17 ...
$ vs : num 0 0 1 1 0 1 0 1 1 1 ...
$ am : num 1 1 1 0 0 0 0 0 0 0 ...
$ gear: num 4 4 4 3 3 3 4 4 4 4 ...
$ carb: num 4 4 1 1 2 1 4 2 2 4 ...
```
To regress the miles per gallon \(\text{mpg}\) on the regressors \(\text{disp}\), \(\text{hp}\) and \(\text{wt}\) we define input as a projection of \texttt{mtcars}:

\[
\texttt{> input <- mtcars[,c("mpg","disp","hp","wt")]} \\
\texttt{> print(head(input))}
\]

\[
\begin{array}{cccc}
\text{mpg} & \text{disp} & \text{hp} & \text{wt} \\
\text{Mazda RX4} & 21.0 & 160 & 110 & 2.620 \\
\text{Mazda RX4 Wag} & 21.0 & 160 & 110 & 2.875 \\
\text{Datsun 710} & 22.8 & 108 & 93 & 2.320 \\
\text{Hornet 4 Drive} & 21.4 & 258 & 110 & 3.215 \\
\text{Hornet Sportabout} & 18.7 & 360 & 175 & 3.440 \\
\text{Valiant} & 18.1 & 225 & 105 & 3.460 \\
\end{array}
\]
- `disp` is the displacement of an engine which is the cumulative volume of all the cylinders that is displaced by the pistons as they move up and down;
- `hp` is the power of an engine;
- `wt` is the weight of the car.
The formula that is the first argument expresses the fact that the mpg is dependent on disp, hp and wt.

```r
> model <- lm(mpg ~ disp+hp+wt, data=input)
> print(model)
```
lm(formula = mpg ~ disp + hp + wt, data = input)

Coefficients:
(Intercept)    disp     hp     wt
  37.105505  -0.000937  -0.031157  -3.800891

> cat("# # # # The Coefficient Values # # # ","
")
# # # # The Coefficient Values # # #
> a <- coef(model)[1]
> print(a)
(Intercept)
  37.10551
> Xdisp <- coef(model)[2]
> Xhp <- coef(model)[3]
> Xwt <- coef(model)[4]
>
> print(Xdisp)
disp
-0.0009370091