Notes on
"Least-Squares"
Fitting Lines and Other Curves to Data

(This is the part of Section 4.4 on which you have an assignment)

The situation is: You have done some experiment or measurement and have a collection of data points. (We will work just with 2-dimensional data.) The coordinates represent a relationship between two variables and the data set seems to lie approximately on a straight line or other curve. Your understanding of the physical situation should guide you in deciding what kind of relationship is being observed: linear? quadratic? exponential? sinusoidal? The method of least squares is a general approach to finding the curve of a particular type that "best" fits the given data. I put the word "best" in quotes because there may be more than one way to define "best". The approach here (and in the text) is probably the most standard/common/familiar one. (Or perhaps, as suggested in class yesterday, I should call this the second-most standard one; the most would be just eyeballing.)

The general idea is: We want to find a function $f(x)$ so that our data is well approximated by the curve $y=f(x)$. The function $f$ is given in terms of some parameters, e.g. $L(x) = mx+b$, or $Q(x) = ax^2 + bx + c$, and our goal is to find good values for the parameters (for example $m$ and $b$ in the line situation, coefficients $a$, $b$, $c$ in the quadratic situation).

Here are three examples. We will start with a data set, and fit three kinds of curves: a line, a parabola, and an exponential function.

```plaintext
> MyData:={[1,2], [2,3], [3,3], [4,6], [5,8]};
   MyData := {[2, 3], [3, 3], [4, 6], [5, 8]}
> with(plots):
> DrawPoints:=pointplot(MyData, color=red, symbol=box):
> display(DrawPoints, view=[0..6, 0..8], scaling=constrained);
```
First, let's find the "least-squares line" for this data set.

We define a function $L(x) = mx + b$, where $m$ and $b$ are unknown parameters yet to be found. The whole analysis will be devoted to finding the "best" $m$ and $b$.

You are used to seeing "$x$" and "$y$" as the variables of interest; but here "$m$" and "$b$" are what we are trying to find.

We will write a function $E(m,b)$ that measures how good or bad is the line $y=mx+b$ as an approximation of the given data.

```latex
> L:=x\rightarrow m*x+b;
```

$L := x \rightarrow m \times x + b$

For each of the five data points in the set MyData, calculate the difference between the observed $y$-value and the value that would be predicted by the function $L$.

The first data point is [1,2]. So here the observed $y$-value is 2. The $y$-value that the line $L$ would predict is $L(1)=m*1+b$. 

Similarly, the difference for the second data point [2,3] is 3 - L(2), which is

> E2:=3-L(2);

\[ E2 := 3 - 2m - b \]

> E3:=3-L(3);

\[ E3 := 3 - 3m - b \]

> E4:=6-L(4);

\[ E4 := 6 - 4m - b \]

> E5:=8-L(5);

\[ E5 := 8 - 5m - b \]

We now square each of these individual errors and add them up, to get a measure of the total error involved in using \( y=mx+b \) to estimate the data points.

> Err:=E1^2 + E2^2 + E3^2 + E4^2 + E5^2;

\[ Err := (2 - m - b)^2 + (3 - 2m - b)^2 + (3 - 3m - b)^2 + (6 - 4m - b)^2 + (8 - 5m - b)^2 \]

This \( Err \) is a function of 2 variables, \( m \) and \( b \). We want to find values for \( m \) and \( b \) that make this expression minimum. We do this by finding where \( dErr/dm = 0 = dErr/db \).

> dEdm:=diff(Err,m);

\[ dEdm := -162 + 110m + 30b \]

> dEdb:=diff(Err,b);

\[ dEdb := -44 + 30m + 10b \]

Remark: You can calculate the partial derivatives by first squaring the terms in \( Err \) and collecting the \( m \) and \( b \) terms to get a "tidy" quadratic in \( m \) and \( b \), or you can differentiate without first expanding \( Err \). The latter might be easier, and is closer to how one analyzes this method theoretically: \( Err = (2-m-b)^2 + (3-2m-b)^2 + \text{etc} \Rightarrow dErrdm = 2(2-m-b)(-1) + 2(3-2m-b)(-2) + \text{etc} \).

> Equations:={dEdm=0, dEdb=0};

\[ \text{Equations} := \{-162 + 110m + 30b = 0, -44 + 30m + 10b = 0\} \]

Notice this is a linear system of two equations in two unknowns, easy for you to solve.

> BestMB:=solve(Equations,{m,b});

\[ \text{BestMB} := \left\{ m = \frac{3}{2}, b = \frac{-1}{10} \right\} \]

There is only one critical point, and we know (from thinking about how the \( Err \) function is defined, e.g always positive) that this must correspond to a global minimum. So if a person defines "best" as "minimize the expression \( Err \)", then the "best" linear function \( y=L(x) \) that approximates the given data is

> Lbest:=x->(3/2)*x+(-1/10);
Now let's fit a quadratic \( y = ax^2 + bx + c \) to the same data set.

We define a function \( Q(x) = ax^2 + bx + c \), where \( a, b, c \) are unknown parameters yet to be found. The whole analysis will be devoted to finding the "best" \( a, b, c \).

You are used to seeing "\( x \)" and "\( y \)" as the variables of interest; but here \( a, b, c \) are what we are trying to find. We will write a function \( E(a, b, c) \) that measures how good or bad is the parabola \( y = ax^2 + bx + c \) as an approximation of the given data.

> \( Q := x \rightarrow a \cdot x^2 + b \cdot x + c \);
\[ Q := x \rightarrow ax^2 + bx + c \]

For each of the five data points in the set MyData, calculate the difference between the observed y-value and the value that would be predicted by the function Q.

The first data point is [1,2]. So here the observed y-value is 2. The y-value that the line Q would predict is \( Q(1) = a*1^2 + b*1 + c \).

\[
\begin{align*}
> & \text{E1:=2-Q(1);} \\
& E1 := 2 - a - b - c
\end{align*}
\]

Similarly, the difference for the second data point [2,3] is 3 - Q(2), which is

\[
\begin{align*}
> & \text{E2:=3-Q(2);} \\
& E2 := 3 - 4a - 2b - c
\end{align*}
\]

\[
\begin{align*}
> & \text{E3:=3-Q(3);} \\
& E3 := 3 - 9a - 3b - c
\end{align*}
\]

\[
\begin{align*}
> & \text{E4:=6-Q(4);} \\
& E4 := 6 - 16a - 4b - c
\end{align*}
\]

\[
\begin{align*}
> & \text{E5:=8-Q(5);} \\
& E5 := 8 - 25a - 5b - c
\end{align*}
\]

We now square each of these individual errors and add them up, to get a measure of the total error involved in using \( y=mx+b \) to estimate the data points.

\[
\begin{align*}
> & \text{Err:=E1^2 + E2^2 + E3^2 + E4^2 + E5^2;} \\
& \text{Err := \((2-a-b-c)^2 + (3-4a-2b-c)^2 + (3-9a-3b-c)^2 + (6-16a-4b-c)^2 + (8-25a-5b-c)^2\)}
\end{align*}
\]

This \( \text{Err} \) is a function of 2 variables, \( m \) and \( b \). We want to find values for \( m \) and \( b \) that make this expression minimum. We do this by finding where \( \text{dErr/dm = 0 = dErr/db} \).

\[
\begin{align*}
> & \text{dEda:=diff(Err,a);} \\
& \text{dEda := -674 + 1958a + 450b + 110c}
\end{align*}
\]

\[
\begin{align*}
> & \text{dEdb:=diff(Err,b);} \\
& \text{dEdb := -162 + 450a + 110b + 30c}
\end{align*}
\]

\[
\begin{align*}
> & \text{dEdc:=diff(Err,c);} \\
& \text{dEdc := -44 + 110a + 30b + 10c}
\end{align*}
\]

Remark: You can calculate the partial derivatives by first squaring the terms in \( \text{Err} \) and collecting the \( a,b,c \) terms to get a "tidy" quadratic in \( a,b,c \), or you can differentiate without first expanding \( \text{Err} \). The latter might be easier, and is closer to how one analyzes this method theoretically: \( \text{Err} = (2-a-b-c)^2 + (3-4a-2b-c)^2 + etc. \implies \text{dE/da = 2(2-a-b-c)(-1) + 2(3-4a-2b-c)(-4) + etc.} \)

\[
\begin{align*}
> & \text{Equations:={dEda=0, dEdb=0, dEdc=0};} \\
& \text{Equations := \{-674 + 1958a + 450b + 110c = 0, -44 + 110a + 30b + 10c = 0, -162 + 450a + 110b + 30c = 0\}}
\end{align*}
\]
Notice this is a linear system of three equations in three unknowns, easy (?) - at least you know you can do it) for you to solve.

```maple
Bestabc := solve(Equations, {a, b, c});
Bestabc := \{ a = \frac{5}{14}, b = -\frac{9}{14}, c = \frac{12}{5} \}
```

```maple
Qbest := x -> (5/14)*x^2 + (-9/14)*x + (12/5);
Qbest := x \rightarrow \frac{5}{14}x^2 - \frac{9}{14}x + \frac{12}{5}
```

```maple
DrawPara := plot(Qbest(x), x=0..6, color=blue):
DisplayPara := display({DrawPoints, DrawPara}, view=[0..6, 0..8], scaling=constrained);
```

Finally, let's try to fit an exponential function to the given data.

We define a function \( F(x) = A e^{rx} \), where \( A \) and \( r \) are unknown parameters yet to be found. The whole analysis will be devoted to finding the "best" \( A \) and \( r \). (We also could add an option of a vertical translation and
look for \( F(x) = Ae^{rx} + b \)

You are used to seeing "x" and "y" as the variables of interest; but here \( A \) and \( r \) are what we are trying to find.

We will write a function \( F(A, r) \) that measures how good or bad is the parabola \( y = Ae^{rx} \) as an approximation of the given data.

\[
F := x \rightarrow Ae^{rx}
\]

For each of the five data points in the set \( \text{MyData} \), calculate the difference between the observed \( y \)-value and the value that would be predicted by the function \( F \).

\[
E1 := 2 - F(1);
\]

\[
E1 := 2 - Ae^r
\]

Similarly, the difference for the second data point \([2, 3]\) is \( 3 - F(2) \), which is

\[
E2 := 3 - F(2);
\]

\[
E2 := 3 - Ae^{2r}
\]

\[
E3 := 3 - F(3);
\]

\[
E3 := 3 - Ae^{3r}
\]

\[
E4 := 6 - F(4);
\]

\[
E4 := 6 - Ae^{4r}
\]

\[
E5 := 8 - F(5);
\]

\[
E5 := 8 - Ae^{5r}
\]

We now square each of these individual errors and add them up, to get a measure of the total error involved in using \( y = mx + b \) to estimate the data points.

\[
\text{Err} := E1^2 + E2^2 + E3^2 + E4^2 + E5^2;
\]

\[
\text{Err} := (2 - Ae^r)^2 + (3 - Ae^{2r})^2 + (3 - Ae^{3r})^2 + (6 - Ae^{4r})^2 + (8 - Ae^{5r})^2
\]

This \( \text{Err} \) is a function of 2 variables, \( m \) and \( b \). We want to find values for \( m \) and \( b \) that make this expression minimum. We do this by finding where \( d\text{Err}/dm = 0 = d\text{Err}/db \).

\[
d\text{EdA} := \text{diff(Err, A)};
\]

\[
d\text{EdA} := -2(2 - Ae^r)e^r - 2(3 - Ae^{2r})e^{2r} - 2(3 - Ae^{3r})e^{3r} - 2(6 - Ae^{4r})e^{4r} - 2(8 - Ae^{5r})e^{5r} - 2(2 - Ae^r)e^r - 2(3 - Ae^{2r})e^{2r} - 2(3 - Ae^{3r})e^{3r} - 2(6 - Ae^{4r})e^{4r} - 2(8 - Ae^{5r})e^{5r}
\]

\[
d\text{Edr} := \text{diff(Err, r)};
\]

\[
d\text{Edr} := -2(2 - Ae^r)Ae^r - 4(3 - Ae^{2r})Ae^{2r} - 6(3 - Ae^{3r})Ae^{3r} - 8(6 - Ae^{4r})Ae^{4r} - 2(2 - Ae^r)Ae^r - 4(3 - Ae^{2r})Ae^{2r} - 6(3 - Ae^{3r})Ae^{3r} - 8(6 - Ae^{4r})Ae^{4r}
\]
- 10 (8 - A e^{(5 r)}) A e^{(5 r)}$

Remark: Maple has chosen to differentiate without first expanding Err. Before trying to calculate the critical point (A,r), let's tell the computer to collect terms.

\[
\begin{align*}
\text{dEdA} & := \text{collect(expand(dEdA),A)}; \\
\text{dEdA} & := \left( 2 (e^r)^2 + 2 (e^r)^4 + 2 (e^r)^6 + 2 (e^r)^8 + 2 (e^r)^{10} \right) A - 4 (e^r)^2 - 6 (e^r)^3 - 12 (e^r)^4 - 16 (e^r)^5 \\
\text{dEdr} & := \text{collect(expand(dEdr),A)}; \\
\text{dEdr} & := \left( 2 (e^r)^2 + 4 (e^r)^4 + 6 (e^r)^6 + 8 (e^r)^8 + 10 (e^r)^{10} \right) A^2 \\
& \quad + \left( -4 e^r - 12 (e^r)^2 - 18 (e^r)^3 - 48 (e^r)^4 - 80 (e^r)^5 \right) A \\
\end{align*}
\]

\[
\text{Equations} := \{ \text{dEdA} = 0, \text{dEdr} = 0 \};
\]

\[
\begin{align*}
\text{Equations} & := \left\{ \begin{align*}
2 (e^r)^2 + 2 (e^r)^4 + 2 (e^r)^6 + 2 (e^r)^8 + 2 (e^r)^{10} \right) A & - 4 (e^r)^2 - 6 (e^r)^3 - 12 (e^r)^4 - 16 (e^r)^5 = 0, \\
2 (e^r)^2 + 4 (e^r)^4 + 6 (e^r)^6 + 8 (e^r)^8 + 10 (e^r)^{10} \right) A^2 & + \left( -4 e^r - 12 (e^r)^2 - 18 (e^r)^3 - 48 (e^r)^4 - 80 (e^r)^5 \right) A = 0 \end{align*} \right\}
\]

We can simplify these equations by dividing through by $e^{(2r)}$, but it still looks messy. Let's just ask for numerical solutions.

\[
\text{BestAr} := \text{fsolve(Equations,\{A,r\})};
\]

\[
\text{BestAr} := \text{fsolve}\left\{ \begin{align*}
2 (e^r)^2 + 2 (e^r)^4 + 2 (e^r)^6 + 2 (e^r)^8 + 2 (e^r)^{10} \right) A & - 4 (e^r)^2 - 6 (e^r)^3 - 12 (e^r)^4 - 16 (e^r)^5 = 0, \\
2 (e^r)^2 + 4 (e^r)^4 + 6 (e^r)^6 + 8 (e^r)^8 + 10 (e^r)^{10} \right) A^2 & + \left( -4 e^r - 12 (e^r)^2 - 18 (e^r)^3 - 48 (e^r)^4 - 80 (e^r)^5 \right) A = 0 \end{align*} \right\}, \{A, r\}
\]

OK. Maple is still stumped. Let's try again for numerical solution, but give Maple a hint about where to start looking.

\[
\text{BestAr} := \text{fsolve(Equations,\{A=1,r=1\})};
\]

\[
\text{BestAr} := \{ A = 1.263887753, r = 0.3704301354 \}
\]

\[
\text{ExpBest} := x \rightarrow 1.263887753 \times \exp(0.3704301354 \times x);
\]
\[ \text{ExpBest} := x \rightarrow 1.263887753 e^{0.3704301354 x} \]

> DrawExp:=plot(ExpBest(x), x=0..6, color=blue):
> display({DrawPoints, DrawExp}, view=[0..6, 0..8], scaling=constrained);

Final Comment: Which kind of curve is the "best" to use for the given data?? There are two philosophical approaches to this question. One is to consider a lot of possible curves (within some physically sensible family - we all know we can fit a 99 degree polynomial exactly to 100 data points, but that does not seem an enlightening way to model) and find where the error function Err is smallest. Or a person can say, "I know [or want to test the hypothesis that] from the physics of this situation, the relationship between the variables x and y that I am studying must be an exponential one of this particular form, so I will try to fit an exponential function of that form. There has been an interesting debate on this in modeling how the human body responds to external stimuli. We measure sound in decibels because it seems like each 10 times increase in physical intensity produces approximately a one "unit" increase in how people perceive the loudness: is this a logarithmic relation? or a fractional power relation? For interesting discussion of such issues see <http://en.wikipedia.org/wiki/Weber-Fechner_law> and the link there to the "Stevens power law". 