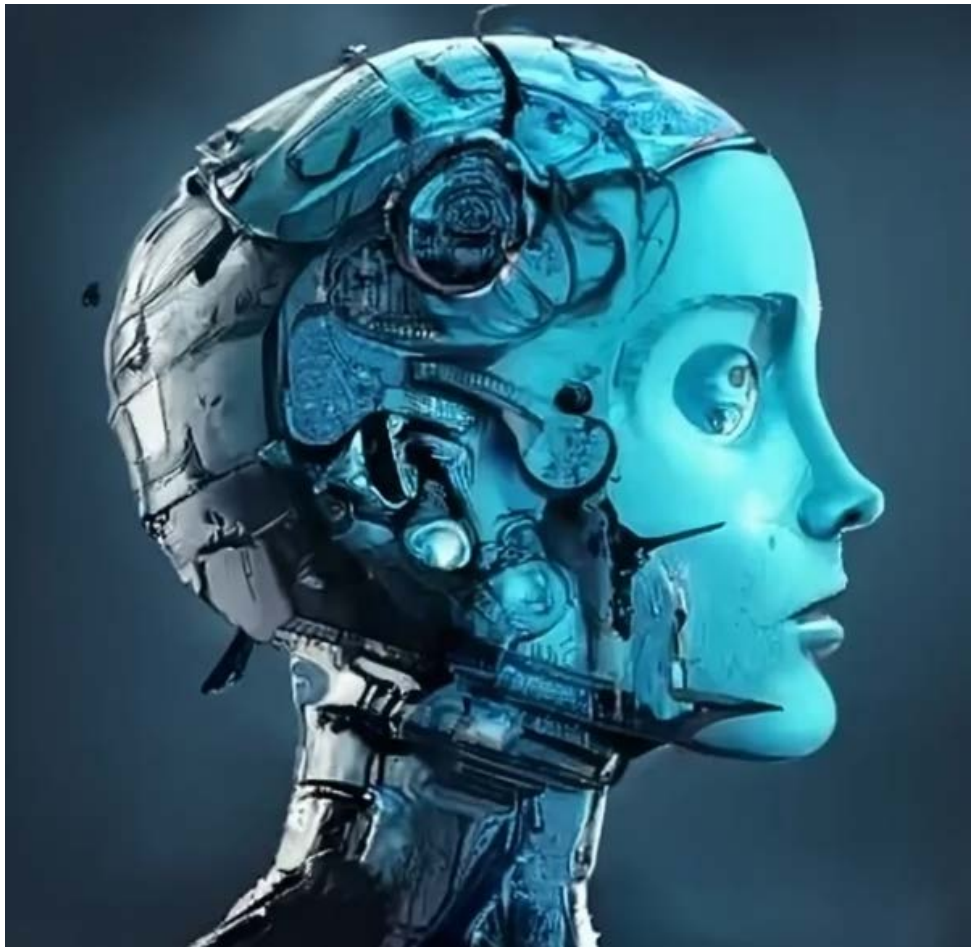
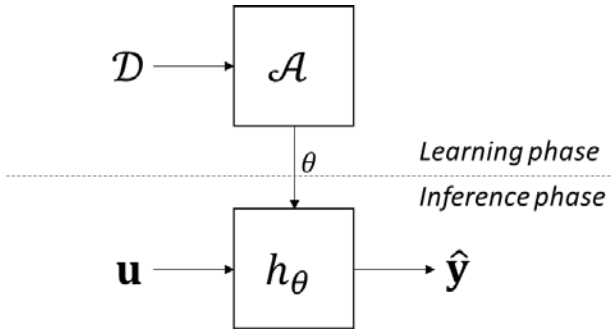
Machine Intelligence

Linear Regression

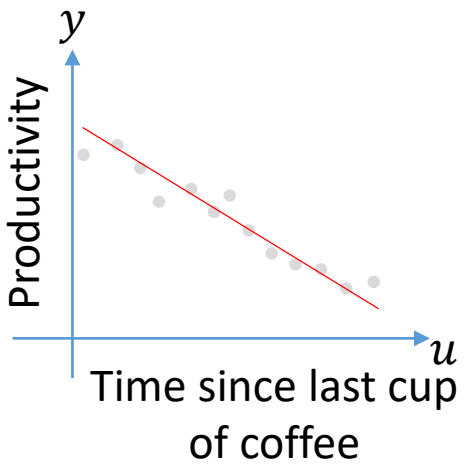
Dr. Cory Merkel



Linear Regression in 2D



- A.K.A. fitting a straight line to data



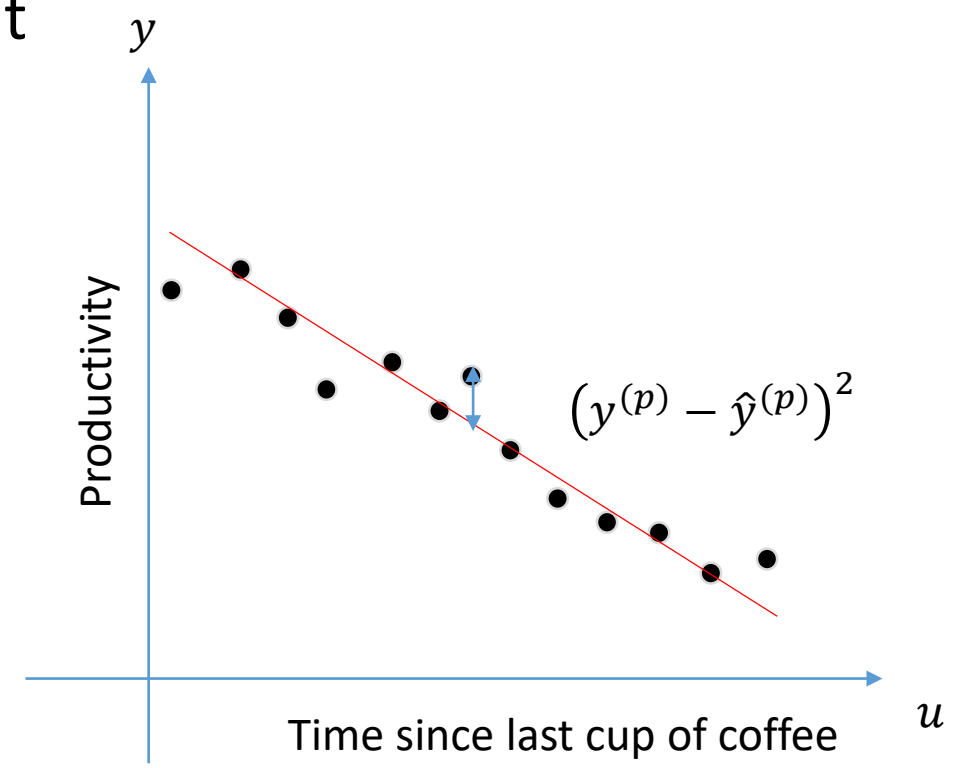
- $\hat{y} = h_\theta(u) = mu + b, \theta = \{m, b\}$
- $\mathcal{D} = \left\{\left(u^{(p)}, y^{(p)}\right)\right\}$
  - We will call the number of samples in our dataset n
- What is $\mathcal{A}$???

Linear Regression in 2D

- Before we can define $\mathcal{A}$, we need to know what our goal is
- To fit a straight line to this data, the goal that makes the most sense is to minimize the average squared difference between each point and the line

$$\mathcal{L} = \frac{1}{n}\sum_{p=1}^{n}\left(y^{(p)} - \hat{y}^{(p)}\right)^2$$

- We call this the *loss function*
  - n is the size of our dataset, and p is an index
  - This particular loss function is the mean squared error loss (MSE)



Linear Regression in 2D

- Let's expand our loss:

$$\mathcal{L} = \frac{1}{n}\sum_{p=1}^{n}\left(y^{(p)} - \hat{y}^{(p)}\right)^2 = \frac{1}{n}\sum_{p=1}^{n}\left(y^{(p)} - \left[mu^{(p)} + b\right]\right)^2$$

- Our goal is to minimize $\mathcal{L}$, so take the derivative with respect to both parameters and set the results equal to 0

$$\frac{\partial \mathcal{L}}{\partial m} = \frac{2}{n}\sum_{p=1}^{n}(y^{(p)} - mu^{(p)} - b)u^{(p)} = 0$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{2}{n}\sum_{p=1}^{n}(y^{(p)} - mu^{(p)} - b) = 0$$

- Now, we have 2 equations and 2 unknowns, so we can solve

Linear Regression in 2D

- Start with the 2nd equation and solve for b:

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{2}{n}\sum_{p=1}^{n}(y^{(p)} - mu^{(p)} - b) = 0$$

$$b = \frac{1}{n}\sum_{p=1}^{n} y^{(p)} - \frac{m}{n}\sum_{p=1}^{n} u^{(p)}$$

- Plug this into the first equation and solve for m:

$$\frac{\partial \mathcal{L}}{\partial m} = \frac{2}{n}\sum_{p=1}^{n}\left(y^{(p)} - mu^{(p)} - \left[\frac{1}{n}\sum_{p=1}^{n} y^{(p)} - \frac{m}{n}\sum_{p=1}^{n} u^{(p)}\right]\right) u^{(p)} = 0$$

$$= \sum_{p=1}^{n} u^{(p)}y^{(p)} - m\sum_{p=1}^{n}\left(u^{(p)}\right)^2 - \frac{1}{n}\sum_{p=1}^{n} y^{(p)}\sum_{p=1}^{n} u^{(p)} + \frac{m}{n}\left(\sum_{p=1}^{n} u^{(p)}\right)^2$$

$$\rightarrow \quad m = \frac{n\sum_{p=1}^{n} u^{(p)}y^{(p)} - \sum_{p=1}^{n} y^{(p)}\sum_{p=1}^{n} u^{(p)}}{\sum_{p=1}^{n}(u^{(p)})^2 - \left(\sum_{p=1}^{n} u^{(p)}\right)^2}$$

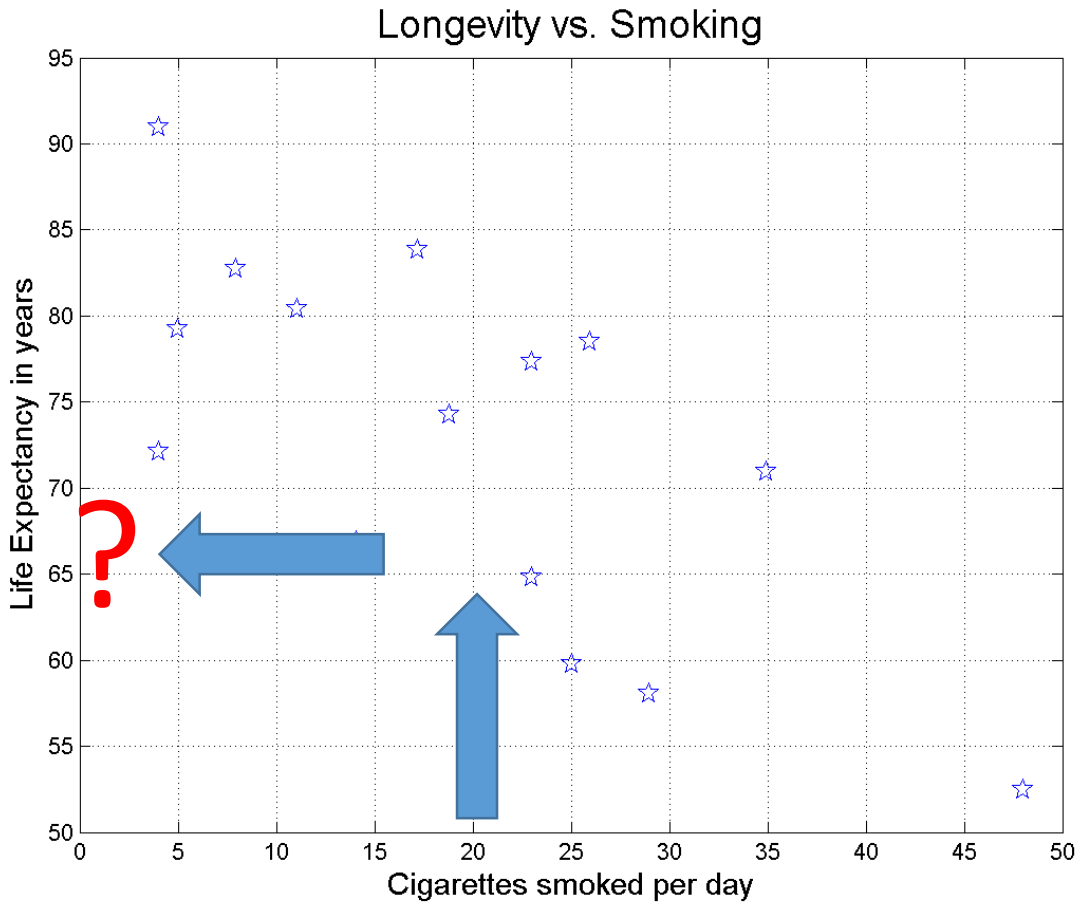
Linear Regression in 2D

- Simplifying the notation, we can finally write

$$m = \frac{n\sum uy - \sum u \sum y}{n\sum u^2 - (\sum u)^2}$$

$$b = \frac{1}{n}\sum y - \frac{m}{n}\sum u$$

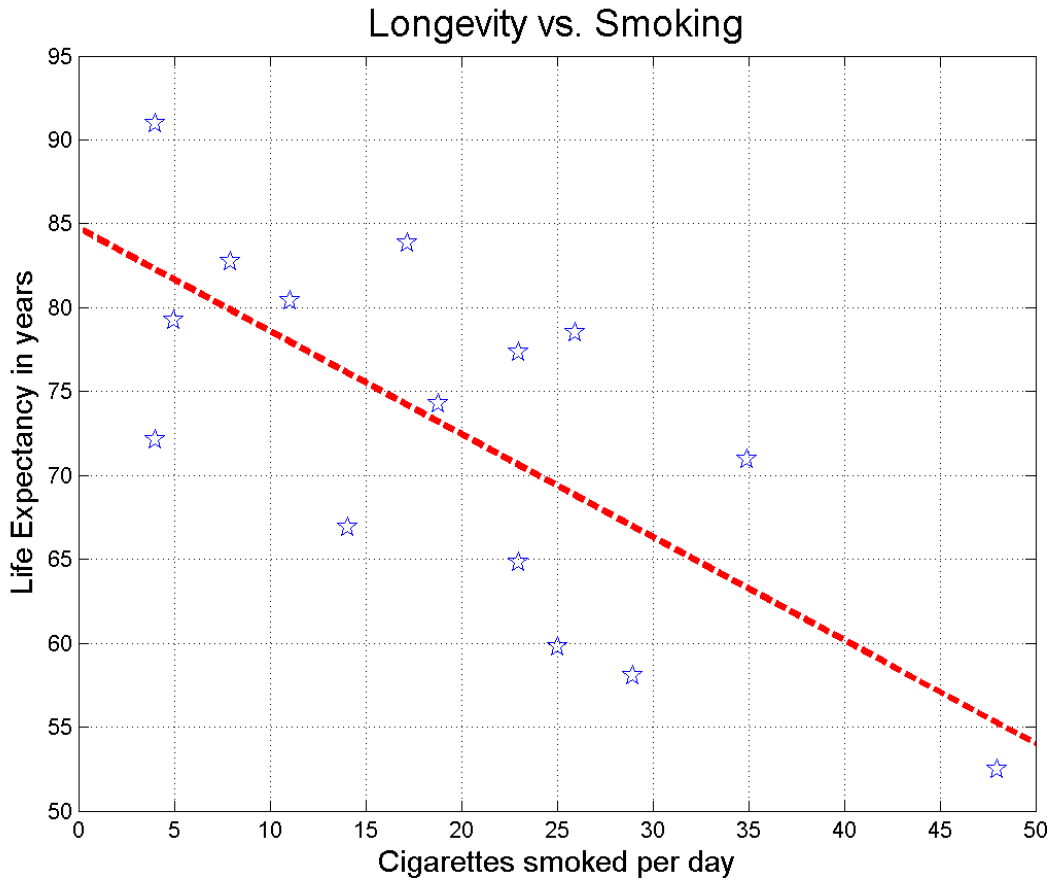
Let's Try it Out on Some Real Data



$\mathcal{D}$

u	y	uu	uy
4.00	91.00	16.00	364.00
4.98	79.26	24.81	394.81
4.00	72.14	16.00	288.54
7.93	82.72	62.82	655.63
11.03	80.41	121.74	887.23
14.06	66.94	197.67	941.11
17.17	83.88	294.72	1439.95
18.80	74.25	353.55	1396.17
22.97	77.33	527.80	1776.64
22.97	64.82	527.80	1489.17
25.02	59.81	625.93	1496.49
25.92	78.49	671.75	2034.25
28.94	58.08	837.77	1681.15
34.91	70.98	1219.02	2478.23
48.00	52.50	2304.00	2520.00
Σu	Σy	Σu^2	Σuy
290.71	1092.61	7801.39	19843.38

Let's Try it Out on Some Real Data



$\mathcal{D}$

u	y	uu	uy
4.00	91.00	16.00	364.00
4.98	79.26	24.81	394.81
4.00	72.14	16.00	288.54
7.93	82.72	62.82	655.63
11.03	80.41	121.74	887.23
14.06	66.94	197.67	941.11
17.17	83.88	294.72	1439.95
18.80	74.25	353.55	1396.17
22.97	77.33	527.80	1776.64
22.97	64.82	527.80	1489.17
25.02	59.81	625.93	1496.49
25.92	78.49	671.75	2034.25
28.94	58.08	837.77	1681.15
34.91	70.98	1219.02	2478.23
48.00	52.50	2304.00	2520.00
Σu	Σy	Σu^2	Σuy
290.71	1092.61	7801.39	19843.38

$\boldsymbol{m = -0.6149}$ $\boldsymbol{b = 84.7574}$

Linear Regression in N Dimensions

- In the previous example, we had 1 independent variable and 1 dependent variable.
  - What if we have more independent variables?

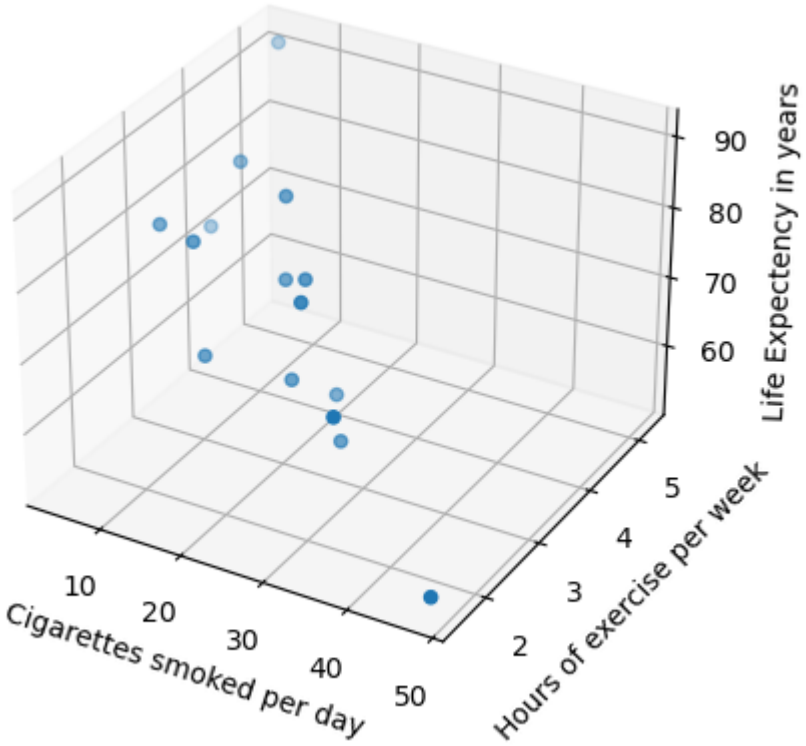
$\mathbf{u}^{(1)}$ → $u_1^{(1)}$, $u_2^{(1)}$; $\mathbf{u}^{(4)}$

$\mathbf{U} =$

Cigarettes smoked per day	Hours of exercise per week
4.00	5.25
4.98	3
4.00	4
7.93	4
11.03	2.75
14.06	2.5
17.17	3.5
18.80	3.25
22.97	3
22.97	2.75
25.02	3.25
25.92	2.5
28.94	2.75
34.91	1.75
48.00	1.5

$\mathbf{y} =$

Life Expectancy
91.00
79.26
72.14
82.72
80.41
66.94
83.88
74.25
77.33
64.82
59.81
78.49
58.08
70.98
52.50



Linear Regression in N Dimensions

- In N dimensions, our hypothesis function becomes

$$\hat{y}^{(p)} = h\left(\mathbf{u}^{(p)}\right) = u_1^{(p)} w_1 + u_2^{(p)} w_2 + \cdots + u_N^{(p)} w_N + w_o$$

  - Note that we switched from m and b to w_i $(\theta = \{w_i\})$
  - This is still a *linear* model, but we have a linear term corresponding to each component or *feature* of the input
    - ❑This is also the equation of a hyperplane!

- Let's write this in matrix form:

$$\mathbf{w} = [w_1, w_2, \ldots, w_N]^T$$

$$\hat{\mathbf{y}} = \mathbf{U}\mathbf{w} + w_o$$

- If our model works perfectly for the data that we have observed, then

$$\mathbf{y} = \mathbf{U}\mathbf{w} + w_o$$

No "hat"

These are the inputs and outputs from my dataset…

Augmenting the Data Matrix

- This is a nice compact form, but it would be better if we could absorb w_o into $\mathbf{w}$
- To get w_o with the rest of the parameters, we can augment the data matrix like this:

$$\mathbf{U} = \left[\mathbf{1}|\mathbf{U}_{\mathrm{og}}\right]$$

where $\mathbf{U}_{\mathrm{og}}$ is the original matrix and $\mathbf{1}$ is a column vector of ones

- Now, $\mathbf{w} = [w_0, w_1, w_2, \ldots, w_N]$, and $\hat{\mathbf{y}} = \mathbf{U}\mathbf{w}$

Augmented Matrix Example

- For our smoking example, the augmented data matrix would look like:

$$\mathbf{U} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \textbf{Cigarettes smoked per day} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \textbf{Hours of exercise per week} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{bmatrix} 1 & 4.00 & 5.25 \\ 1 & 4.98 & 3 \\ 1 & 4.00 & 4 \\ 1 & 7.93 & 4 \\ 1 & 11.03 & 2.75 \\ 1 & 14.06 & 2.5 \\ 1 & 17.17 & 3.5 \\ 1 & 18.80 & 3.25 \\ 1 & 22.97 & 3 \\ 1 & 22.97 & 2.75 \\ 1 & 25.02 & 3.25 \\ 1 & 25.92 & 2.5 \\ 1 & 28.94 & 2.75 \\ 1 & 34.91 & 1.75 \\ 1 & 48.00 & 1.5 \end{bmatrix}$$

Matrix Methods for Linear Regression

- Now we have this equation: $\mathbf{y} = \mathbf{U}\mathbf{w}$, or $\mathbf{U}\mathbf{w} = \mathbf{y}$
- How do we find $\mathbf{w}$?
- It would be nice if we could do this:

Identity matrix:

$$\begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$

$$\mathbf{U}\mathbf{w} = \mathbf{y}$$
$$\mathbf{U}^{-1}\mathbf{U}\mathbf{w} = \mathbf{U}^{-1}\mathbf{y}$$
$$\mathbf{I}\mathbf{w} = \mathbf{U}^{-1}\mathbf{y}$$
$$\mathbf{w} = \mathbf{U}^{-1}\mathbf{y}$$

- But, in general, $\mathbf{U}$ will not be invertible…. ☹

Matrix Methods for Linear Regression

- $\mathbf{U}^{-1}$ will not generally exist, but if the columns of $\mathbf{U}$ are linearly independent, then $\mathbf{U}^T\mathbf{U}$ has an inverse
- Outline of proof: Consider $\mathbf{Ux} = \mathbf{0}$
  - If $\mathbf{U}$ has independent columns, then the only solution is $\mathbf{x} = \mathbf{0}$
    - Recall that the space of solutions to this equation is called the null space
  - Now, show that:
    - If $\mathbf{x}$ is in the null space of $\mathbf{U}$, it is also in the null space of $\mathbf{U}^{\mathrm{T}}\mathbf{U}$
    - If $\mathbf{x}$ is in the null space of $\mathbf{U}^{\mathrm{T}}\mathbf{U}$, it is also in the space of $\mathbf{U}$
  - This tells us that the only solution to $\mathbf{U}^{\mathrm{T}}\mathbf{Ux} = \mathbf{0}$ is $\mathbf{x} = \mathbf{0}$ and therefore $\mathbf{U}^{\mathrm{T}}\mathbf{U}$ is invertible!

Matrix Methods for Linear Regression

- Since $\mathbf{U^T U}$ is invertible, we can do this:

$$\mathbf{Uw} = \mathbf{y}$$
$$(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{Uw} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{y}$$
$$\mathbf{Iw} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{y}$$
$$\mathbf{w} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{y}$$

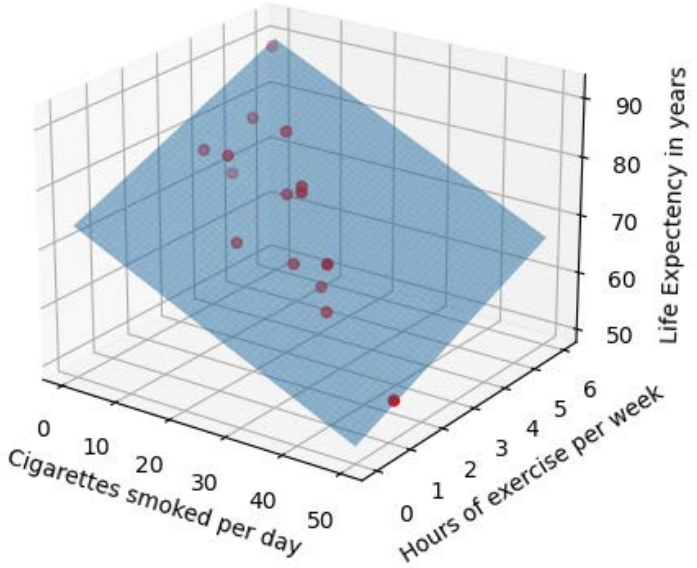
- Notice that $(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T$ multiplied by $\mathbf{U}$ gives the identify matrix
  - Therefore, we call this term a *pseudoinverse* of $\mathbf{U}$ (the Moore-Penrose pseudoinverse in this case)
- Sometimes, we may write $\mathbf{U}^\dagger \equiv (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T$, so

$$\mathbf{w} = \mathbf{U}^\dagger\mathbf{y}$$

Example

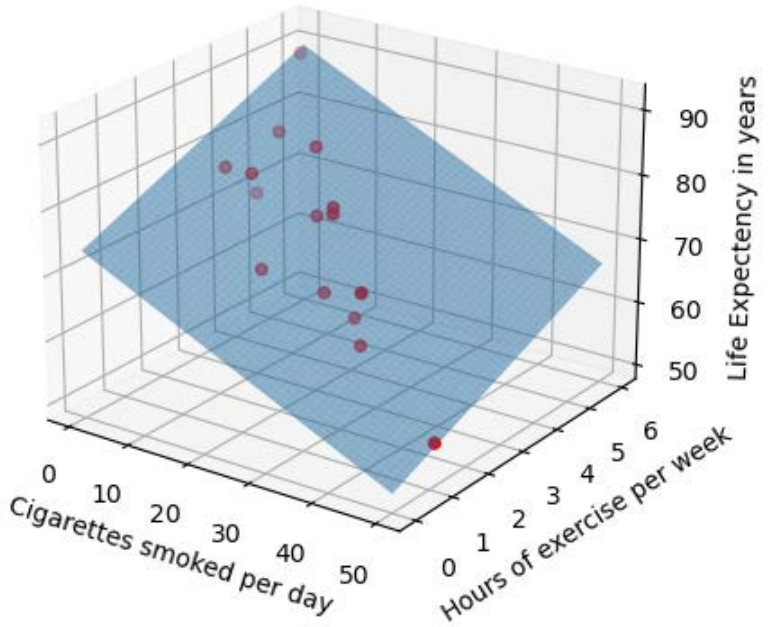
- Applying this to our smoking example, we get: $\mathbf{w} = \mathbf{U}^{\dagger}\mathbf{y} = [73.7, -0.46, 2.65]$

$$Life\ expectency = 73.7 - 0.46 \times \frac{Cigarettes}{day} + 2.65 \times \frac{exercise}{week}$$



But Wait…

- We started our derivation with this: $\mathbf{y} = \mathbf{U}\mathbf{w}$, but looking at our plot, we see that our datapoints don't fit the plane exactly ($\mathbf{y}^{(p)} \neq \hat{\mathbf{y}}^{(p)}$)
  - So what have we actually done?
  - We've actually found the solution that minimizes the MSE loss (just as in the 2D case!)



MSE Loss in N Dimensions

- For the 2D case, we had $\mathcal{L} = \frac{1}{n}\sum_{p=1}^{n}\left(y^{(p)} - \hat{y}^{(p)}\right)^2$
- For N dimensions we have the same thing:

$$\mathcal{L} = \frac{1}{n}\sum_{p=1}^{n}\left(y^{(p)} - \hat{y}^{(p)}\right)^2 = (\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}}) = \left(\mathbf{y}^T - \hat{\mathbf{y}}^{\mathrm{T}}\right)(\mathbf{y} - \hat{\mathbf{y}})$$

$$= \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\hat{\mathbf{y}} - \hat{\mathbf{y}}^T\mathbf{y} + \hat{\mathbf{y}}^T\hat{\mathbf{y}} = \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{U}\mathbf{w} - (\mathbf{U}\mathbf{w})^T\mathbf{y} + (\mathbf{U}\mathbf{w})^T\mathbf{U}\mathbf{w}$$

- Just as before, we can set the derivative equal to 0:

$$\underbrace{\frac{\mathrm{d}\mathbf{y}^T\mathbf{y}}{\mathrm{d}\mathbf{w}}}_{0} - \underbrace{\frac{\mathrm{d}\mathbf{y}^T\mathbf{U}\mathbf{w}}{\mathrm{d}\mathbf{w}}}_{-\mathbf{y}^T\mathbf{U}} - \underbrace{\frac{\mathrm{d}(\mathbf{U}\mathbf{w})^T\mathbf{y}}{\mathrm{d}\mathbf{w}}}_{-\mathbf{y}^T\mathbf{U}} + \underbrace{\frac{\mathrm{d}(\mathbf{U}\mathbf{w})^T\mathbf{U}\mathbf{w}}{\mathrm{d}\mathbf{w}}}_{\mathbf{w}^T\mathbf{U}\mathbf{U}^T + \mathbf{w}^T\mathbf{U}^T\mathbf{U}} = \mathbf{0}$$

NOTE: All of these derivatives are the derivative of a scalar w.r.t. a vector. What does this mean???

- So, $\mathbf{U}^T\mathbf{U}\mathbf{w} = \mathbf{U}^T\mathbf{y}$ (Normal equations)

What if we Have More Than One Dependent Variable?

- Imagine now that we want to use cigarettes smoked per day and hours of exercise per week to predict multiple things:
  - Life expectancy
  - Annual healthcare cost
  - …

- We can organize our output data into a matrix with multiple columns

What if we Have More Than One Dependent Variable?

$\mathbf{U} =$

	Cigarettes smoked per day	Hours of exercise per week
$\mathbf{u}^{(1)}$	4.00	5.25
	4.98	3
	4.00	4
$\mathbf{u}^{(4)}$	7.93	4
	11.03	2.75
	14.06	2.5
	17.17	3.5
	18.80	3.25
	22.97	3
	22.97	2.75
	25.02	3.25
	25.92	2.5
	28.94	2.75
	34.91	1.75
	48.00	1.5

$u_1^{(1)}$ $u_2^{(1)}$

$\mathbf{Y} =$

Life Expectancy	Annual Healthcare Cost
91.00	$1,000
79.26	…
72.14	
82.72	
80.41	
66.94	
83.88	
74.25	
77.33	
64.82	
59.81	
78.49	
58.08	
70.98	
52.50	

- Now, we will have 2 linear models
  - One for life expectancy and one for annual healthcare cost

What if we Have More Than One Dependent Variable?

- In general, if we have M dependent variables and N independent variables, we can solve for each of the M linear models simultaneously as

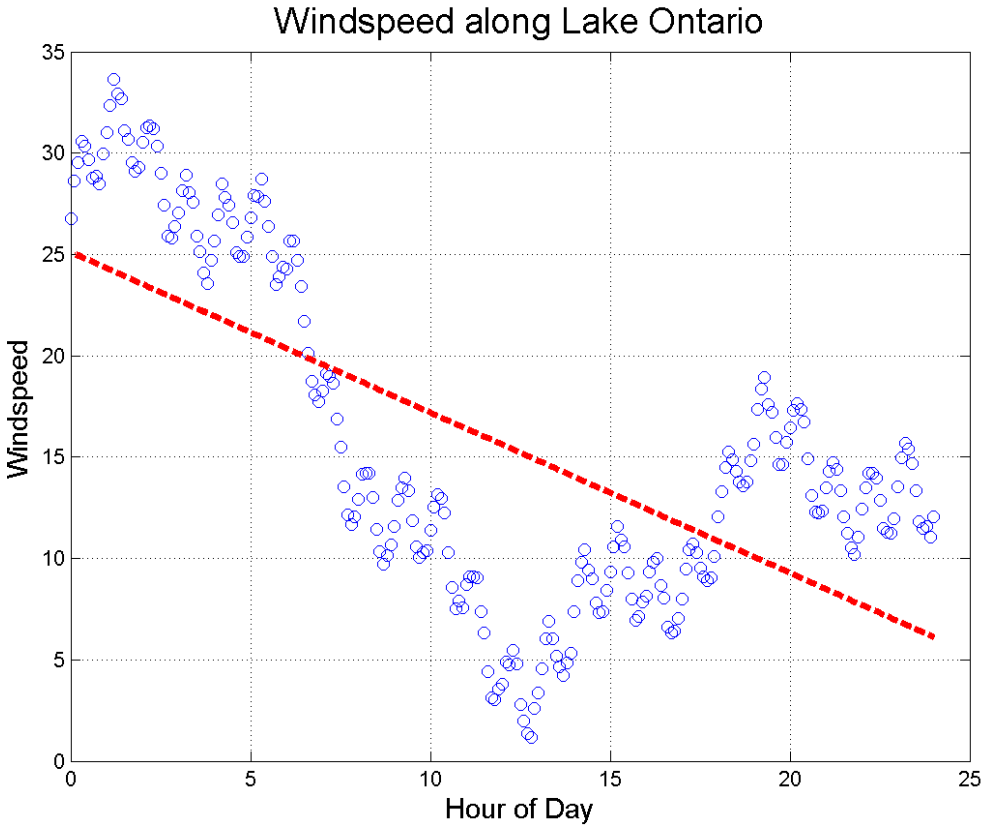
$$\mathbf{W} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{Y}$$

Where $\mathbf{W}$ is an $N \times M$ matrix whose columns correspond to each linear model:

$$\hat{\mathbf{Y}} = \mathbf{UW}$$

Linear Regression with Non-Linear Features

- What if our data are not linear?

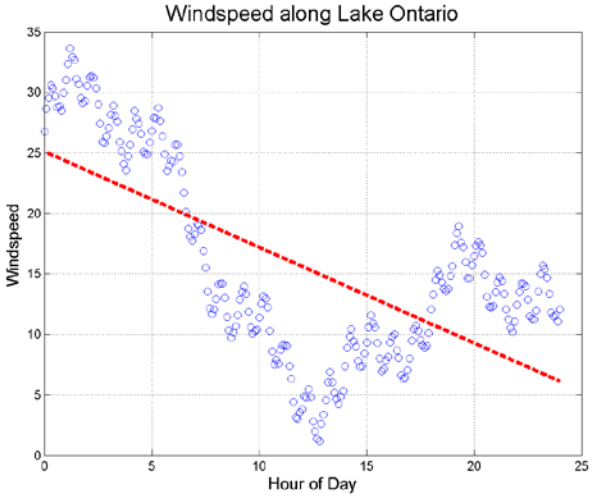


Linear Regression with Non-Linear Features

- We can take our datapoints $\mathbf{u}$ from our dataset and create more data by applying any function to $\mathbf{u}$ that we wish:

We call these *features* of the input

$$\mathbf{u}_{new} = [f_1(\mathbf{u}) \;\; f_2(\mathbf{u}), \ldots,]$$

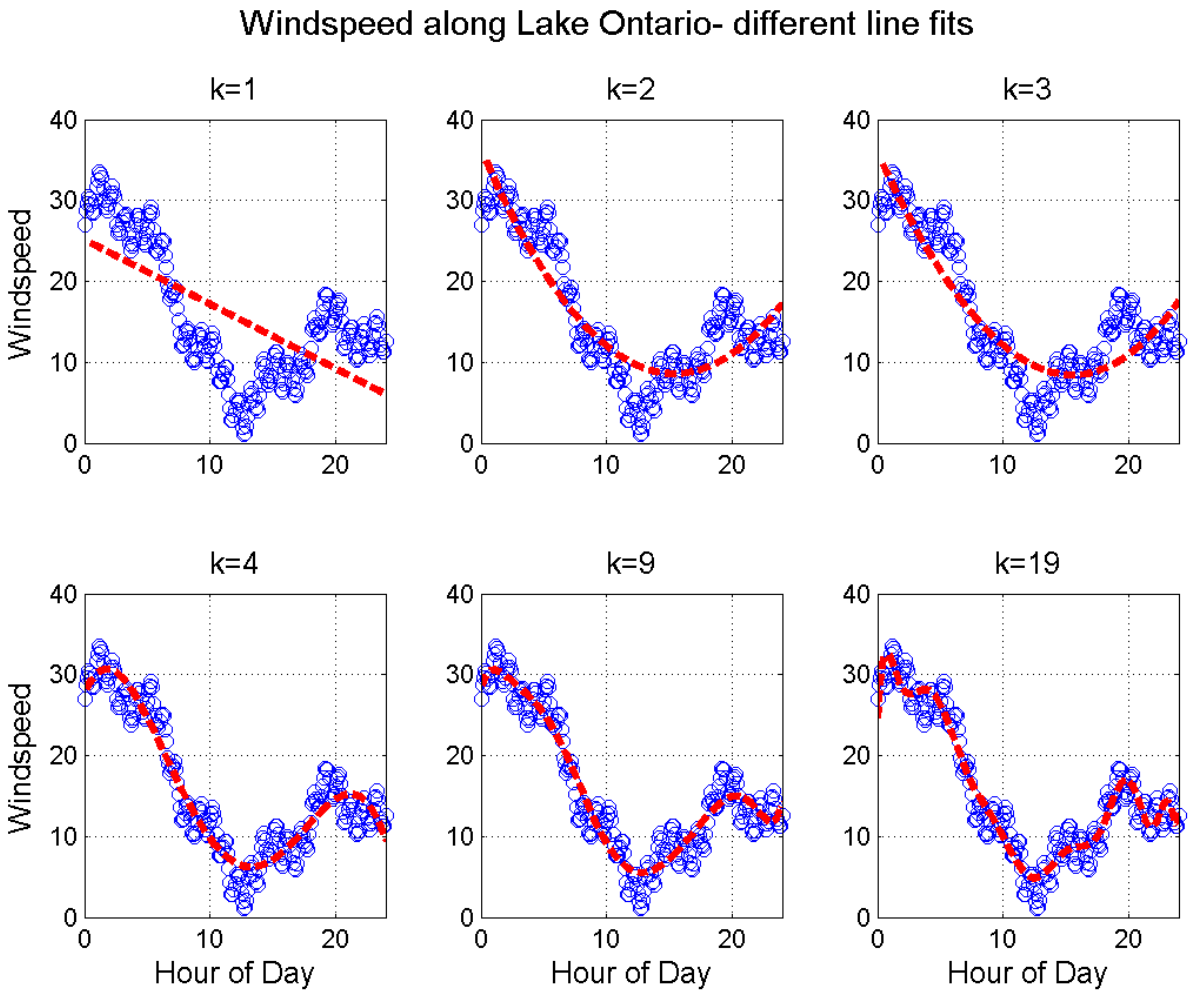


- For example, if our input only has one component, we could take multiple polynomials for our new input in the following way:

$$\mathbf{u}_{new} = [u \;\; u^2 \; u^3 \;\; \ldots \;\; u^k]$$

- This would allow us to fit polynomials to our data
- But there are infinite options, e.g.: $\mathbf{u}_{new} = [e^u u \;\; \log(u) \sin(u) \;\; \ldots \;]$

Linear Regression with Non-Linear Features



Slide based on lecture notes from R. Ptucha, A. Loui, A. Kwasinski, A. Savakis

Why MSE?

- Why did we choose the mean squared error as our loss function?
  - It makes sense intuitively
  - It's *convex*, meaning there is one global optimum

- But, maybe there are other loss functions that would work better?
  - There is a deeper reason for using MSE

Why MSE?

- Recall the idea of finding our model parameters using the log-likelihood:

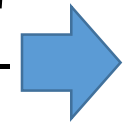
$$\theta_{ML} = \arg\max_{\theta} \sum \log p_h\left(\mathcal{D}^i|\theta\right) = \arg\max_{\theta} \sum \log p_h(\mathbf{y}|\mathbf{u},\mathbf{w})$$

- We will say that the probability of our model outputting $\mathbf{y}$ given the input and the model parameters will be normally distributed:

$$p_h(\mathbf{y}|\mathbf{u},\mathbf{w}) = \mathcal{N}(\mathbf{y};\hat{\mathbf{y}},\sigma^2)$$

- The mean is the model prediction $\hat{\mathbf{y}}$, and the variance σ^2 comes from noise in the model/measurements

$$\sum \log p_h(\mathbf{y}|\mathbf{u},\mathbf{w}) = -n\log\sigma - \frac{n}{2}\log 2\pi - \sum \frac{|\hat{\mathbf{y}}-\mathbf{y}|^2}{2\sigma^2}$$



Note that the maximum of this is found at the same w as the minimum of the MSE: Minimizing MSE corresponds to the maximum likelihood model!