Linear Regression

CS434

Regression analysis

"In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships among variables … focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors'). "

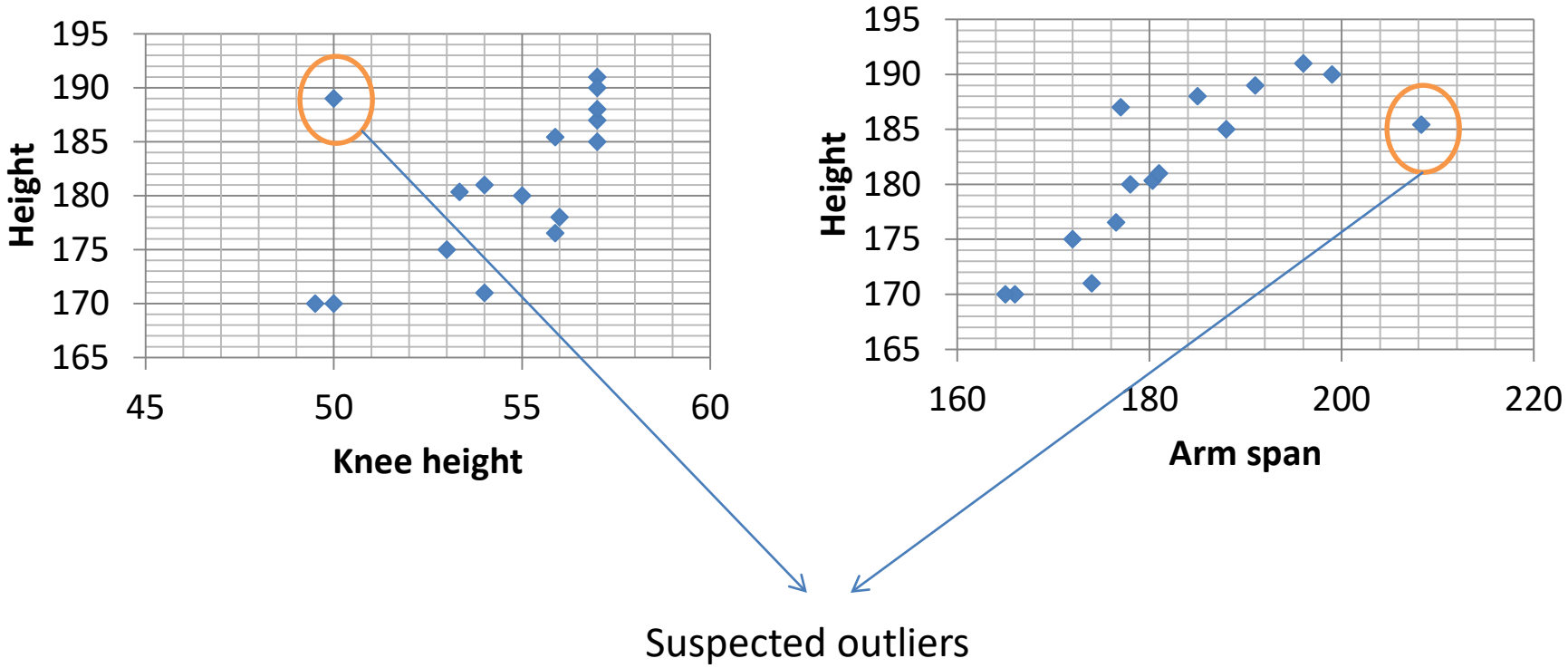
--- Wikipedia

In simple words, we want to predict y (the dependent variable, or **target**) based on a set of x's (independent variables, or **features**), where y is continuous.

A example regression problem

- We want to predict a person's height based on his/her knee height and/or arm span
  - Target: y, the height
  - Features: x_1, the knee height; x_2, the arm span
- This is useful for patients who are bed bound and cannot stand to take an accurate measurement of their height
- Training data:
  - A set of measurements from a subsample of the population

Our Training Data



Suspected outliers

Ignoring these outlier points, there seems to be a reasonable linear relationship between the features and our target variable

Linear prediction function

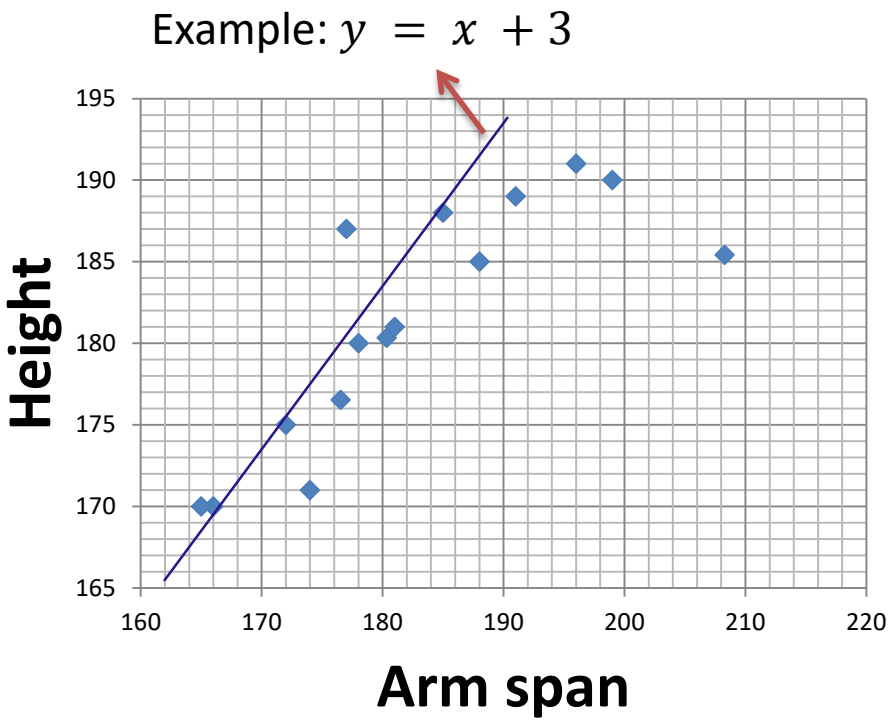
- We will only consider linear functions (thus the name ***linear regression***):

$$y = w_1 x_1 + w_2 x_2 + b$$

- Let's start with just one feature, to make notations simple, we will for now call it x, and our goal is thus to learn a function

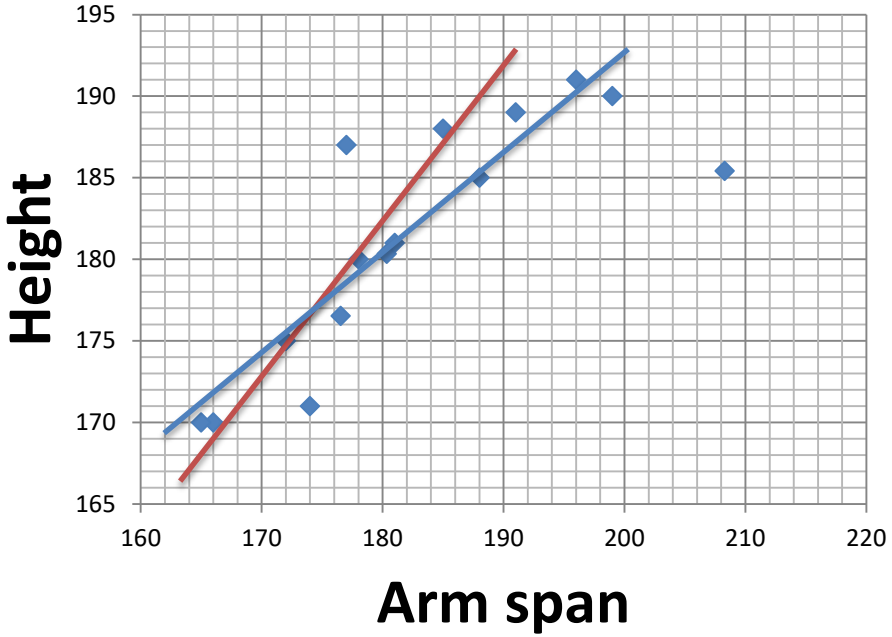
$$y = wx + b$$

One-dimensional Regression



- Goal: fit a line through the points
- Problem: the data does not exactly go through a line

One-dimensional regression



- Which line is better?
- The blue line seems better, but in what way?
- How can we define this goodness precisely?

Let's formalize it a bit more

ID	arm span (x)	Height (y)
1	166	170
2	196	191
3	191	189
4	180.34	180.34
5	174	171
6	176.53	176.53
7	177	187
8	208.28	185.42
9	199	190
10	181	181
11	178	180
12	172	175
13	185	188
14	188	185
15	165	170

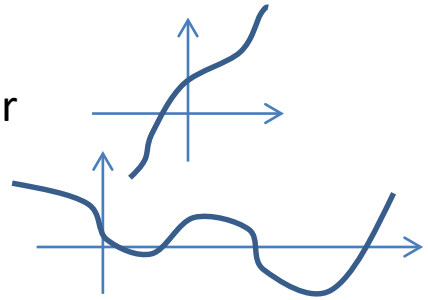
- Given a set of training examples
$$\{(x_i, y_i): i = 1, \dots, n\}$$
- Goal: learn w and b from the training data, so that $y = wx + b$ predicts y_i from x_i accurately
- In mathematical terms, we would like to find the w and b that minimizes the following objective:

$$E(w, b) = \sum_{i=1}^{n} (y_i - (wx_i + b))^2$$

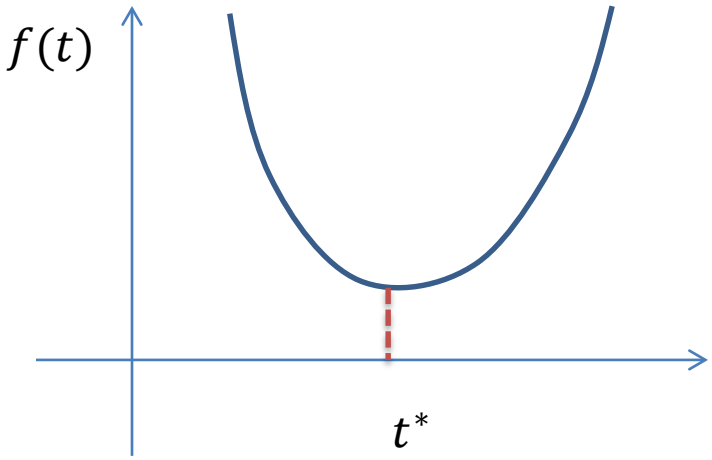
Sum of Squared Error (SSE)

Optimization 101

- Given a function $f(t)$, finding the t^* that minimizes $f(t)$ can be a challenging or impossible in many situations
  - $f(t)$ could be unbounded, without a minimizer
  - $f(t)$ could have a lot of local minimizers



- For the later case, more advanced methods will be needed
- But sometimes, $f(t)$ is well behaved and it is easy to find the optimizer



- In some cases the function is <u>convex</u> (e.g., a simple quadratic objective) and only has one global minimum
- Then it's simple to find the global minimum:
  1. Take derivative of $f(t)$, which we call $f'(t)$
  2. Set it to zero $f'(t) = 0$
  3. Solve for t

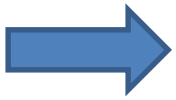
Optimizing $E(w,b) = \sum_{i=1}^{n}(y_i - wx_i - b)^2$

1. Take partial derivative w.r.t. w and b respectively:

$$\frac{\partial E}{\partial w} = \sum_{i=1}^{n} -2(y_i - wx_i - b)x_i ;$$

2. Setting them to zero, and solve for w and b

$$\begin{cases} \frac{\partial E}{\partial b} = \sum_{i=1}^{n} 2(y_i - wx_i - b) = 0 \\ \frac{\partial E}{\partial w} = \sum_{i=1}^{n} 2(y_i x_i - wx_i^2 - bx_i) = 0 \end{cases}$$

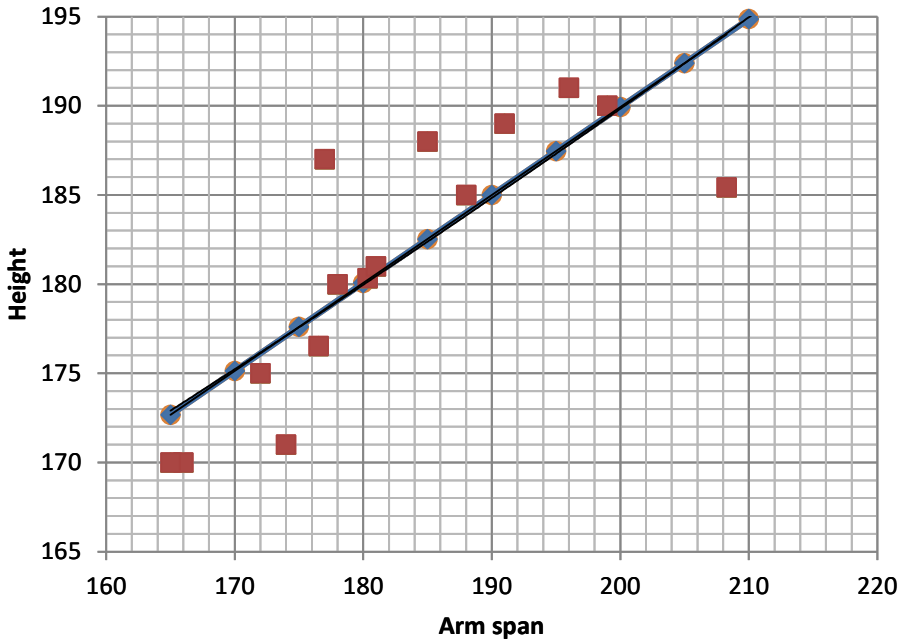


$$b^* = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^* x_i) = \bar{y} - w^* \bar{x}$$

$$w^* = \frac{\overline{yx} - \bar{y}\bar{x}}{\overline{x^2} - \bar{x}^2}$$

Our problem

ID	arm span (x)	Height (y)
1	166	170
2	196	191
3	191	189
4	180.34	180.34
5	174	171
6	176.53	176.53
7	177	187
8	208.28	185.42
9	199	190
10	181	181
11	178	180
12	172	175
13	185	188
14	188	185
15	165	170



$$\bar{x} = \frac{1}{n}\sum_i x_i = 182.477 \qquad \bar{y} = \frac{1}{n}\sum_i y_i = 181.286$$

$$\overline{x^2} = 33436.53 \qquad \overline{xy} = 31148.91 \quad \bar{x}\bar{y} = 33080.46$$

$$w^* = \frac{31148.91 - 33080.46}{33436.53 - 182.477^2} = 0.493$$

$$b^* = 181.3 - 0.493 * 182.5 = 91.30$$

Height = 91.30+0.493*armspan

Extending to more features

- Having more features will mean our objective has more variables to optimize over
- One can solve this similarly by
  - Taking partial derivative of each variable
  - Setting them to zero
  - Solving the system of equations simultaneously

- Use vector calculus, this can be expressed in a succinct way
- Before we do that, we will briefly review some linear algebra and vector calculus notations

Definition: vector

- A **vector** is a one dimensional array.
- We usually denote vectors as boldface lower case letter **x**, and use x to denote a single variable
- If we don't specify otherwise, assume **x** is a column vector

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ \cdots \\ x_{n-1} \end{pmatrix}$$

Adopted from notes by Andrew Rosenberg of CUNY

Definition: matrix

A matrix is a higher dimensional array.
We typically denote matrices as capital letters e.g., A.
If A is an n-by-m matrix, it has the following structure

$$A = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,m-1} \\ a_{1,0} & a_{1,1} & & a_{1,m-1} \\ \vdots & & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,m-1} \end{pmatrix}$$

Adopted from notes by Andrew Rosenberg of CUNY

Transposition

Transposing a matrix or vector swaps rows and columns.

A column-vector becomes a row-vector

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_{n-1} \end{pmatrix}$$

$$\mathbf{x}^T = \begin{pmatrix} x_0 & x_1 & \dots & x_{n-1} \end{pmatrix}$$

Adopted from notes by Andrew Rosenberg of CUNY

Transposing a matrix or vector swaps rows and columns.

A column-vector becomes a row-vector

$$A = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,m-1} \\ a_{1,0} & a_{1,1} & & a_{1,m-1} \\ \vdots & & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,m-1} \end{pmatrix}$$

$$A^T = \begin{pmatrix} a_{0,0} & a_{1,0} & \cdots & a_{n-1,0} \\ a_{0,1} & a_{1,1} & & a_{1,m-1} \\ \vdots & & \ddots & \vdots \\ a_{0,m-1} & a_{1,m-1} & \cdots & a_{n-1,m-1} \end{pmatrix}$$

If A is n-by-m, then A^T is m-by-n.

Adopted from notes by Andrew Rosenberg of CUNY

Matrices can only be added if they have the same dimension.

$$A+B = \begin{pmatrix} a_{0,0}+b_{0,0} & a_{0,1}+b_{0,1} & \cdots & a_{0,m-1}+b_{0,m-1} \\ a_{1,0}+b_{1,0} & a_{1,1}+b_{1,1} & & a_{1,m-1}+b_{1,m-1} \\ \vdots & & \ddots & \vdots \\ a_{n-1,0}+b_{n-1,0} & a_{n-1,1}+b_{n-1,1} & \cdots & a_{n-1,m-1}+b_{n-1,m-1} \end{pmatrix}$$

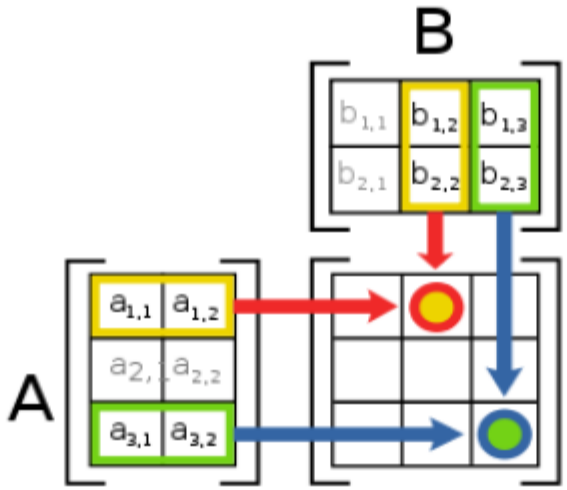
To multiply two matrices, the *inner dimensions* must match.

- An n-by-m can be multiplied by an n'-by-m' matrix iff $m = n'$.

$$AB = C$$

$$c_{ij} = \sum_{k=0}^{m} a_{ik} * b_{kj}$$

That is, multiply the i-th row by the j-th column.



Adopted from notes by Andrew Rosenberg of CUNY

- To multiply two vectors, they must also have their dimensions aligned
- For example:

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = x_1 y_1 + x_2 y_2 + x_3 y_3$$

  - This is often called the inner (or dot) product of two vectors, written as: $\langle \mathbf{x}, \mathbf{y} \rangle$ or $(\mathbf{x} \cdot \mathbf{y}) = \mathbf{x}^T \mathbf{y}$

- Or, alternatively:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & x_1 y_3 \\ x_2 y_1 & x_2 y_2 & x_2 y_3 \\ x_3 y_1 & x_3 y_2 & x_3 y_3 \end{bmatrix}$$

  - This is often called the outer product of two vectors, written as:

$$\mathbf{x} \otimes \mathbf{y} = \mathbf{x}\mathbf{y}^T$$

Useful operations: vector norm

- Given a d-dimensional vector $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$, the (L-2, or Euclidean) norm of $\mathbf{x}$ is represented as

$$\|\mathbf{x}\|_2 = \sqrt[2]{x_1^2 + x_2^2 + \cdots + x_d^2} = \sqrt[2]{< \mathbf{x}, \mathbf{x} >}$$

- There are other norms as well. L_p norm is defined as:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{1/p}$$

Useful operations: Matrix Inversion

- The inverse of a square matrix A is a matrix A^{-1} such that $AA^{-1} = I$, where I is called an identity matrix. For example:

$$A = \begin{bmatrix} 1 & 0 \\ 1 & \frac{1}{2} \end{bmatrix}, A^{-1} = \begin{bmatrix} 1 & 0 \\ -2 & 2 \end{bmatrix}$$

- A square matrix A is invertible iff $|A| \neq 0$, i.e., determinant of A is nonzero
- One way to test if the determinant of A is nonzero if to see if you can come up with a linear combination of the vectors of A so that it equals zero
- if not, we say the columns of A are independent of each other, and $|A| \neq 0$

Some useful Matrix Inversion Properties

$$(A^{-1})^{-1} = A$$

$$(kA)^{-1} = k^{-1}A^{-1}$$

$$(A^T)^{-1} = (A^{-1})^T$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

Multi-dimensional Regression

- Now we will consider the more general case that considers multiple features
- Eg, each example is described by $[x_1 \quad x_2]^T$, where x_1 denotes the arm span, x_2 denotes the knee height
- We want to learn

$$y = w_1 x_1 + w_2 x_2 + b$$

- It is inconvenient to have to represent b separately, so we will use a small trick to represent all the coefficients jointly
  - Include a constant 1 as a dummy input features: $\boldsymbol{x} = [1 \quad x_1 \quad x_2]^T$
  - Let $\boldsymbol{w}$ denote the vector of all coefficients: $\boldsymbol{w} = [b \quad w_1 \quad w_2]^T$
  - The function can be compactly represented as y $= \boldsymbol{w}^T \boldsymbol{x}$

Objective

Previous objective: $$E(w,b) = \sum_{i=1}^{n} (y_i - wx_i - b)^2$$

Updated form $$E(\boldsymbol{w}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (y_i - \boldsymbol{w}^T \boldsymbol{x_i})^{\boldsymbol{2}}$$

- Let $\boldsymbol{y} = [y_1, y_2, \dots, y_n]^T$
- Let $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n]^T$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Example 1

Example n

$$E(\boldsymbol{w}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (y_i - \boldsymbol{w}^T \boldsymbol{x_i})^{\boldsymbol{2}} = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

Optimizing $E(\boldsymbol{w})$

$$E(\boldsymbol{w}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

Take the gradient and setting to zero:

$$\nabla E(\mathbf{w}) = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}_{d+1}$$

d: # of input features

$$\Rightarrow \mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

$$\Rightarrow \mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Matrix cookbook is a good resource to help you with this type of manipulations.

height	knee height	arm span
170	50	166
191	57	196
189	50	191
180.34	53.34	180.34
171	54	174
176.53	55.88	176.53
187	57	177
185.42	55.88	208.28
190	57	199
181	54	181
180	55	178
175	53	172
188	57	185
170	49.5	165
185	57	188

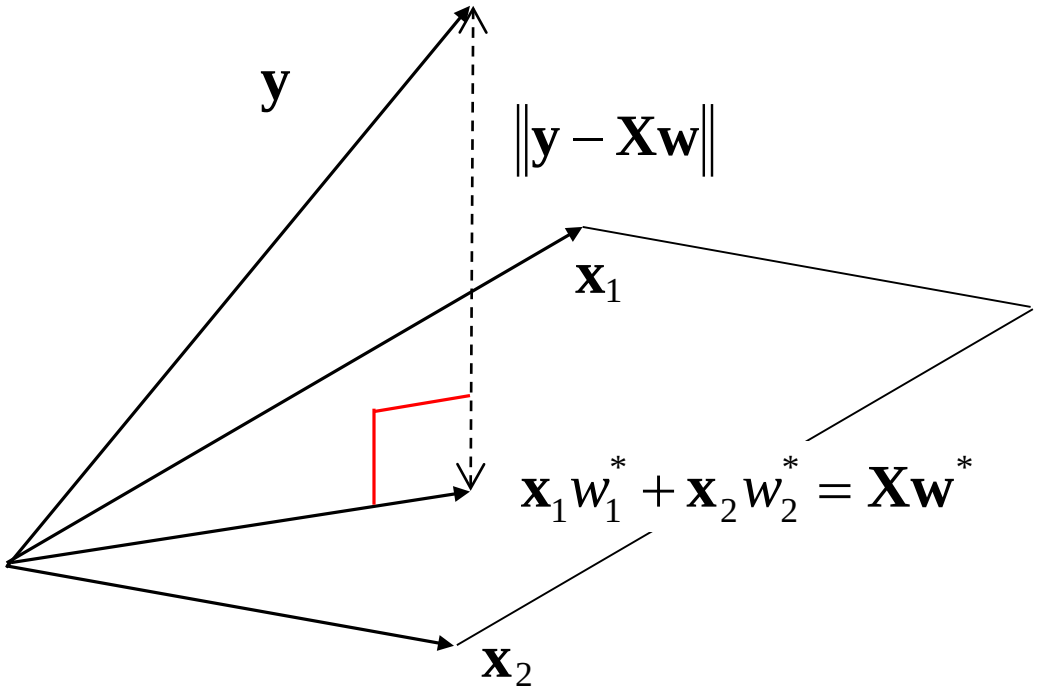
$$X = \begin{bmatrix} 1 & 50 & 166 \\ 1 & 57 & 196 \\ 1 & 50 & 191 \\ 1 & 53.34 & 180.34 \\ 1 & 54 & 174 \\ 1 & 55.88 & 176.53 \\ 1 & 57 & 177 \\ 1 & 55.88 & 208.28 \\ 1 & 57 & 199 \\ 1 & 54 & 181 \\ 1 & 55 & 178 \\ 1 & 53 & 172 \\ 1 & 57 & 185 \\ 1 & 49.5 & 165 \\ 1 & 57 & 188 \end{bmatrix} \qquad Y = \begin{bmatrix} 170 \\ 191 \\ 189 \\ 180.34 \\ 171 \\ 176.53 \\ 187 \\ 185.42 \\ 190 \\ 181 \\ 180 \\ 175 \\ 188 \\ 170 \\ 185 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 15 & 815.6 & 2737.2 \\ 815.6 & 44451.6 & 149081 \\ 2737.2 & 149081 & 501547.9 \end{bmatrix} \qquad \boldsymbol{w} = \begin{bmatrix} 70.19 \\ 0.656 \\ 0.413 \end{bmatrix}$$

Height = 70.19+.656*knee height + 0.413 * arm span

Geometric interpretation

- ***y*** is a vector in R^n
- Each column (feature) of ***X*** is also a vector in R^n
- ***Xw*** is a linear combination of the columns of ***X***, and lies in *range(**X**)*
- We want ***Xw*** to match ***y*** as closely as possible



What is the effect of adding one feature?

- By using both armspan and knee height, can we do better than using just armspan?
- How do we compare?
  - Training SSE with only arm span: 257.445
  - Training SSE with armspan and knee height: 225.680
- Does it mean the model with two features is necessarily better than one?
- More generally, is it always better to have more features?
  - Effect on training?
  - Effect on testing?

Summary

- We introduce linear regression, which assumes that the function that maps from ***x*** to y is linear
- **Sum Squared Error** objective

$$E(\boldsymbol{w}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

- The solution is given by:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- There are other objectives, leading to different solutions
- Although we make the linear assumption, it is easy to use this to learn nonlinear functions as well
  - Introduce nonlinear features, e.g., x_1^2, x_2^2, x_1x_2