Data X

Prediction I

Data, Signals, and Systems

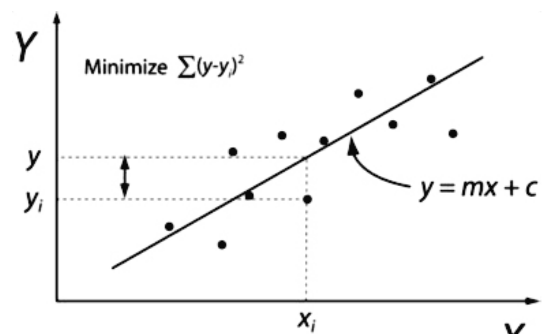
Ikhlaq Sidhu
Chief Scientist & Founding Director,
Sutardja Center for Entrepreneurship & Technology
IEOR Emerging Area Professor Award, UC Berkeley

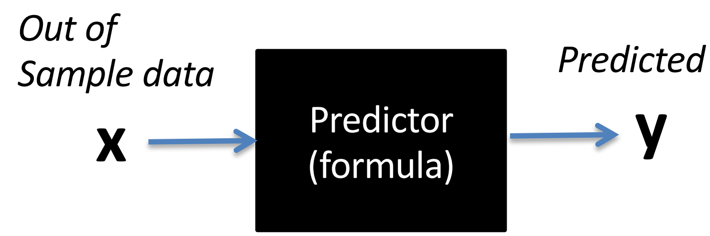
Introduction to Prediction

Prediction

Data We Might Have
(In Sample)

X	*Y*
2	3
5	9
6	11
8	?
10	?
?	?





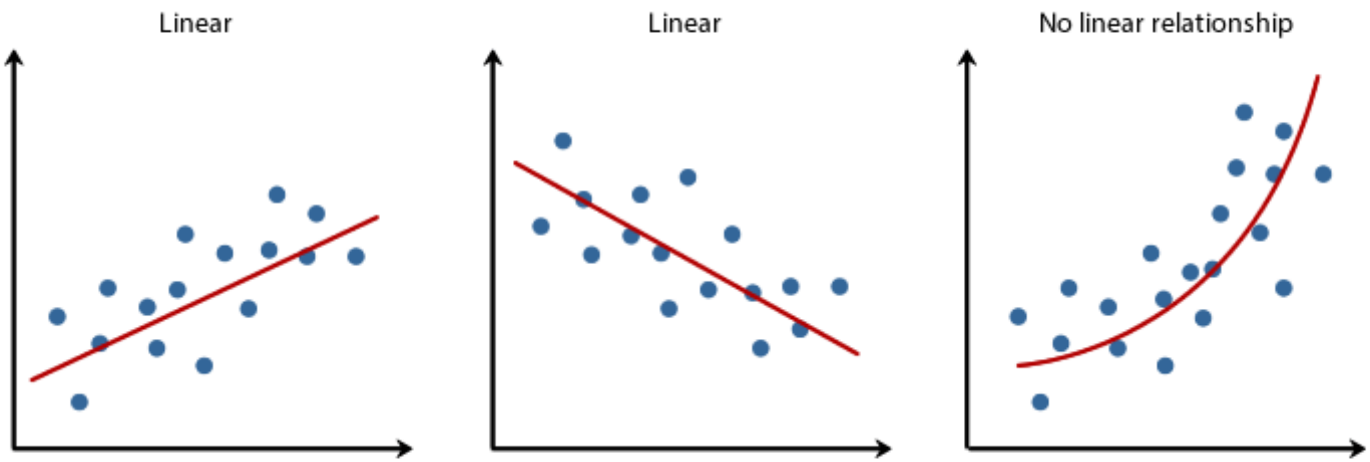
Our Goal: Working with *out of sample data*

Data View

Math View

Systems View

Of Course, we can not assume that all data can be predicted by a ***linear model***



- Model might be a poor fit (wrong model)
- Model might be too good of a fit or over-fit (only works well on the in sample data)

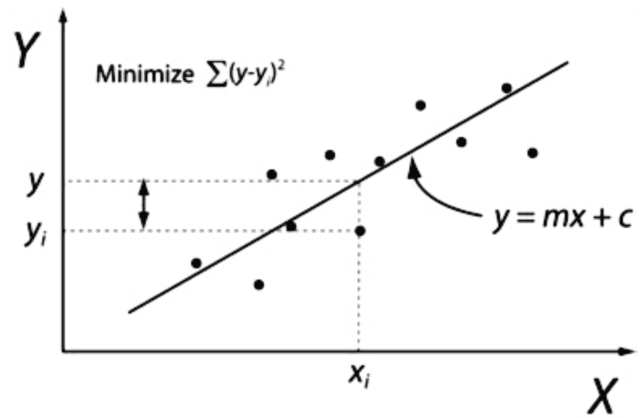
Image: Laerd Statistics, 2014

Prediction

Data We Might Have
(In Sample)

X	Y
2	3
5	9
6	?
8	?
?	?

Data View



One way to make a prediction:
Choose a line that best fits the sample data

Then $\boldsymbol{y(x) = mx + c}$ is a predictor for a new out of sample $\boldsymbol{x}$

Best Linear Predictor
(if you just have 2 Variables)

Remember:
COV(X,Y) = E[(X-μ_x) (Y-μ_y)] = E [X Y] – E[X] E[Y]
COV(X,X) = VAR(X)
COV(AX,Y) = ACOV(X,Y)

This turns out to be
the best linear predictor:

$$L(Y \mid X) = \mathbb{E}(Y) + \frac{\operatorname{cov}(X,Y)}{\operatorname{var}(X)}[X - \mathbb{E}(X)]$$

It's a line:

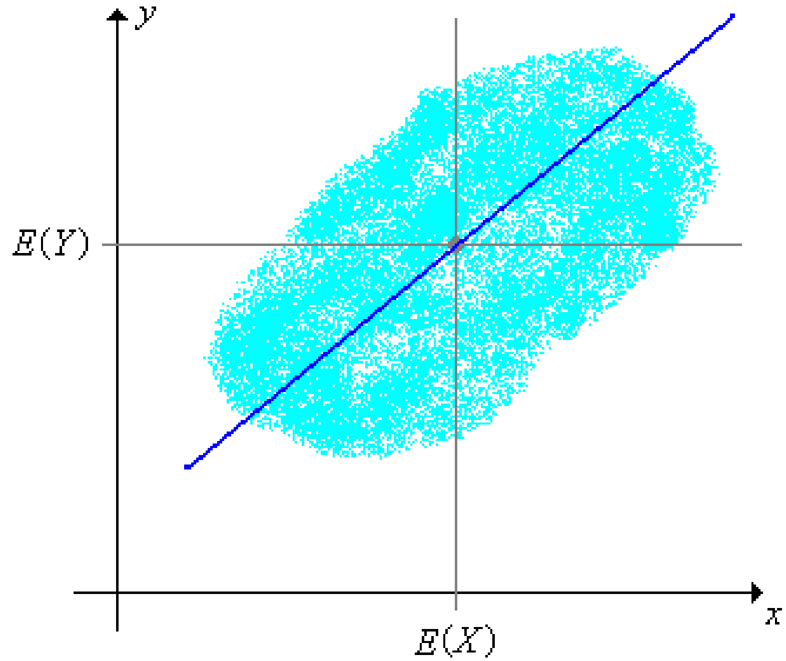
Runs though point: $(E[X],\ E[Y])$

slope: $m = \frac{COV(X,Y)}{VAR(X)}$

y-intercept = $E[Y] - mE[X]$

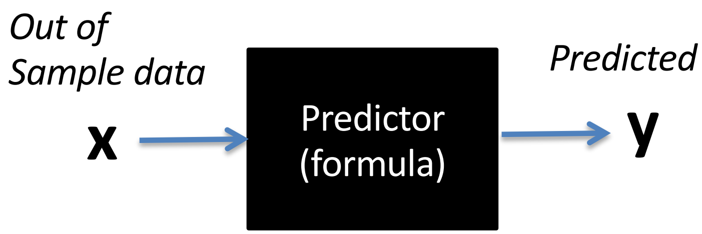
= $E[Y] - COV(X,Y) * \frac{E[X]}{VAR(X)}$

What makes this the the best linear predictor?
How much error does this predictor have?



The distribution regression line

Prediction



$$y\_{estimated} = m\,x + b$$

slope: $m = \frac{COV(X,Y)}{VAR(X)}$

y-intercept $b = E[Y] - mE[X]$

$= E[Y] - COV(X,Y) * \frac{E[X]}{VAR(X)}$

We just learned to estimate y
with a linear predictor for 2 variables (x,y)

Systems View

Try it Yourself:

Data Set in a Table,
two variables

X	Y
2	10
4	5
3	9
5	4
6	3

What would be best linear predictor

Y_estimated = ?

Solution:
What would be best linear predictor?

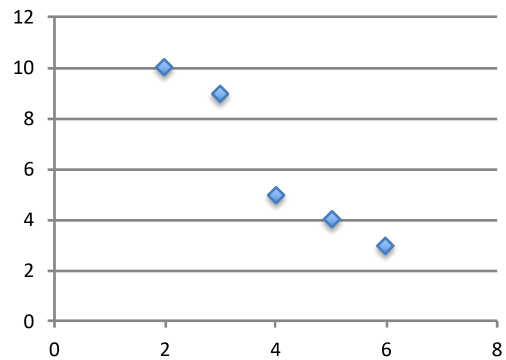
y_estimated = m x + b

Determine slope m, intercept b

What we know already:

X	Y	X*Y	X^2	y(x)
2	10	20	4	6.96
4	5	20	16	3.92
3	9	27	9	5.44
5	4	20	25	2.4
6	3	18	36	0.88

E[X]	E[Y]	E[XY]	E[X^2]
4	6.2	21	18



$$E[X] = 4, \qquad E[Y] = 6.2$$

$$Cov(X,Y) = E[XY] - E[X]E[Y]$$
$$= 21 - 4 * 6.2 = -3.8$$

Var(X) = 18-16 = 2

$$m = \frac{Cov(X,Y)}{Var(X)} = \frac{-3.8}{2} = -1.9$$

y-int $= E[Y] - \frac{Cov(X,Y)}{Var(X)} * E[X]$
$$= 6.2 - \left(\frac{-3.8}{2} * 4\right) = 13.8$$

$$y(x) = -1.9x + 13.8$$

Mean Square Error:
$E([y(x) - y_{actual})^2] =$?

Code Sample

```
import numpy as np

x = np.array([2, 4, 3, 5, 6])
y = np.array([10, 5, 9, 4, 3])

E_x = np.mean(x)
E_y = np.mean(y)

cov_xy = np.mean(x*y)-E_x*E_y

y_0 = E_y - cov_xy/np.var(x)*E_x
m = cov_xy/np.var(x)

y_pred=m*x+y_0

print "E[(y_pred-y_actual)^2] =", np.mean(np.square(y_pred-y))
```

```
E[(y_pred-y_actual)^2] = 0.54
```

Data X

Prediction II

Data, Signals, and Systems

Ikhlaq Sidhu
Chief Scientist & Founding Director,
Sutardja Center for Entrepreneurship & Technology
IEOR Emerging Area Professor Award, UC Berkeley

Prediction

Out of Sample data

x → Predictor (formula) → **y**

Predicted

$$y\_{estimated} = m\,x + b$$

slope: $m = \frac{COV(X,Y)}{VAR(X)}$

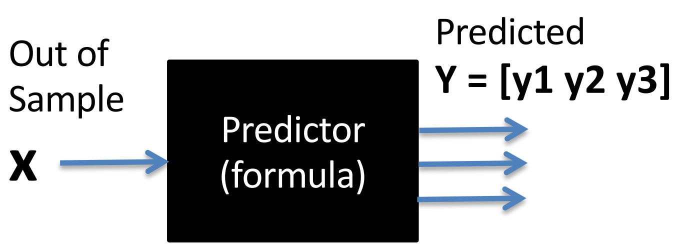
y-intercept $b = E[Y] - mE[X]$

$= E[Y] - COV(X,Y) * \frac{E[X]}{VAR(X)}$

We just learned to estimate y
with a linear predictor for 2 variables (x,y)

Systems View

Prediction: Multiple Inputs and Outputs

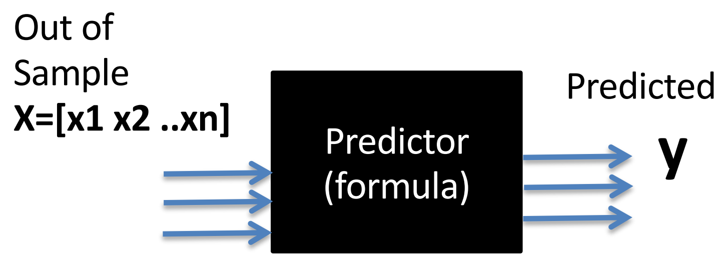


No problem,
we use multiple predictors.

y1 = g1(x)

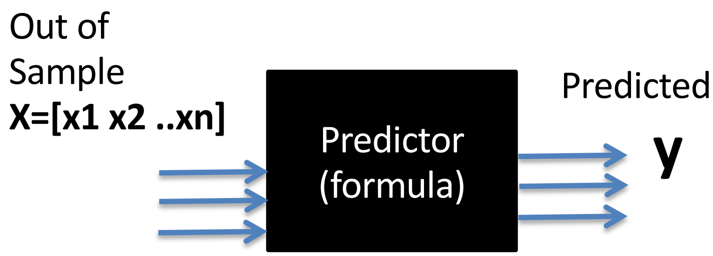
y2 = g2(x)

y3 = g3(x)

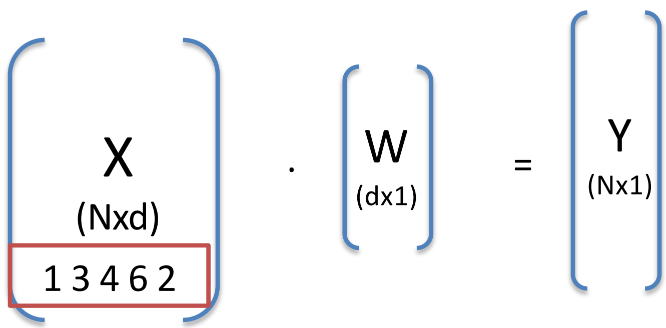


?

Prediction: Multiple Inputs and Outputs



In this case, for linear prediction, we use a matrix format:



$$x_{1,1}w_1 + x_{1,2}w_2 + x_{1,3}w_3 = y_1$$
$$x_{2,1}w_1 + x_{2,2}w_2 + x_{2,3}w_3 = y_2$$

...

...

$\mathbf{x_i}$

X is the in-sample data. We only need to figure out **W**.
With **W**, we can estimate **y** for new **x**, i.e. out of sample data

Example: Prediction with Regression

Data:		**x(i,1)**	**x(i,2)**	**x(i,3)**	**y(i,1)**
ID	Name	**Age**	**years w employer**	**Income**	**Credit Score**
1	John	25	3	50	660
2	Alice	23	2	60	580
3	Bill	28	1	80	425
4	Rahul	25	.5	59	320

$$\underset{(N \times d+1)}{\mathbf{X}} \qquad \mathbf{W} \qquad \underset{N\text{x}1}{\mathbf{Y}}$$

$$\begin{pmatrix} 1 & 25 & 3 & 50 \\ 1 & 23 & 2 & 60 \\ 1 & 28 & 1 & 80 \\ 1 & 25 & .5 & 59 \end{pmatrix} \cdot \begin{pmatrix} w0 \\ w1 \\ w2 \\ w3 \end{pmatrix} = \begin{pmatrix} 660 \\ 580 \\ 425 \\ 320 \end{pmatrix}$$

We are now going to *choose a W that gives us a predictor* for **Y**

This time, **Y** is the actual value we want to estimate

Notice: we added an extra column of 1s?

Example: Prediction with Regression

$$
\begin{array}{c c}
 & \mathbf{X} \\
 & (N \times d{+}1) \\
 & \\
 & 1\ 25\ 3\ 50 \\
\mathbf{X_2} & \boxed{1\ 23\ 2\ 60} \\
 & 1\ 28\ 1\ 80 \\
 & 1\ 25\ .5\ 59
\end{array}
\cdot
\begin{array}{c}
W \\
\begin{pmatrix} w0 \\ w1 \\ w2 \\ w3 \end{pmatrix}
\end{array}
=
\begin{array}{c}
\mathbf{Y} \\
Nx1 \\
\begin{pmatrix} 660 \\ 580 \\ 425 \\ 320 \end{pmatrix}
\end{array}
$$

This time, **Y** is the actual value we want to estimate.?

Train with this to “calculate” **W**

Out of sample
(a row of numbers)

$\mathbf{X_i}$

→ **Predictor:** g(x) = XW →

Estimated y
(a real number)

Then predict (or classify) with **W**

The Math: Linear Regression

Predictor:
g(x) = XW

OK, but better to measure squared error

$$E_{in}(w) = E[Xw - Y]$$

$$E_{in}(w) = \frac{1}{N}\sum_{i=1}^{N}\left(w^T x_i - y_i\right)^2$$

$$E_{in}(w) = \frac{1}{N} \left\| Xw - y \right\|^2$$

$$\nabla E_{in}(w) = \frac{2}{N} X^T (Xw - y) = 0.$$

$$X^T X W = X^T Y$$

$$\boxed{W = (X^T X)^{-1} X^T Y}$$

$$Y_{\text{estimated}} = X_{\text{out of sample}} W$$

X^T = dxN

X = Nxd

$(X^T X)$ = dxd

$(X^T X)^{-1}$ = dxd

$W = (X^T X)^{-1} X^T Y$=

= dxd · dxN · Y= Nxm outputs

= [dxN] x [Nxm] = dxm

Y estimated = X W =

Xout = Nout x d · W = dxm = Nout x m

N = # of X data rows
d= # of features
m= # of outputs in Y

Prediction with Regression (continued)

$$X = \underset{(N \times d+1)}{\begin{pmatrix} 1 & 25 & 3 & 50 \\ 1 & 23 & 2 & 60 \\ 1 & 28 & 1 & 80 \\ 1 & 25 & .5 & 59 \end{pmatrix}} \cdot \underset{\mathbf{W}}{\begin{pmatrix} w0 \\ w1 \\ w2 \\ w3 \end{pmatrix}} = \underset{N\times 1}{\begin{pmatrix} 660 \\ 580 \\ 425 \\ 320 \end{pmatrix}}$$

X (N x d+1) **W** **Y** Nx1

This time, **Y** is the actual value we want to estimate.?

Train with this to "calculate" **W**

$$\boxed{W = X^{\dagger} Y = (X^T X)^{-1} X^T Y}$$

Out of sample (a row of numbers) $\mathbf{X}_i$ → **Predictor:** g(x) = XW → **Estimated y** (a real number)

Then predict (or classify) with **W**

Code Sample

```python
import numpy as np

x = np.array([
        [1,25,3,50],
        [1,23,2,60],
        [1,28,1,80],
        [1,25,0.5,59]
    ])
y = np.array([660,580,425,320])

print "W = ", np.linalg.inv(x.T.dot(x)).dot(x.T).dot(y)
```

```
W =  [ 426.17283951  -16.04938272  149.44444444    3.7345679 ]
```

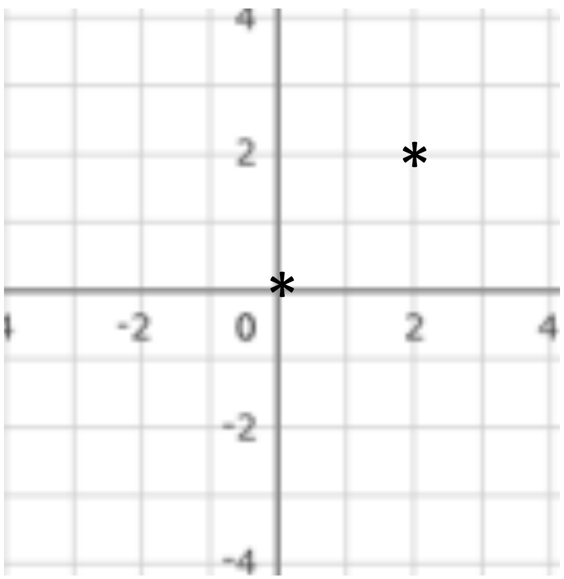
$$x_{0,}w_0 + x_{1,}w_1 + x_2w_2 + x_3w_3 = y_i$$

Data:		x(i,1)	x(i,2)	x(i,3)	y(i,1)
ID	Name	**Age**	**years w employer**	**Income**	**Credit Score**
1	John	25	3	50	660
2	Alice	23	2	60	580
3	Bill	28	1	80	425
4	Rahul	25	.5	59	320

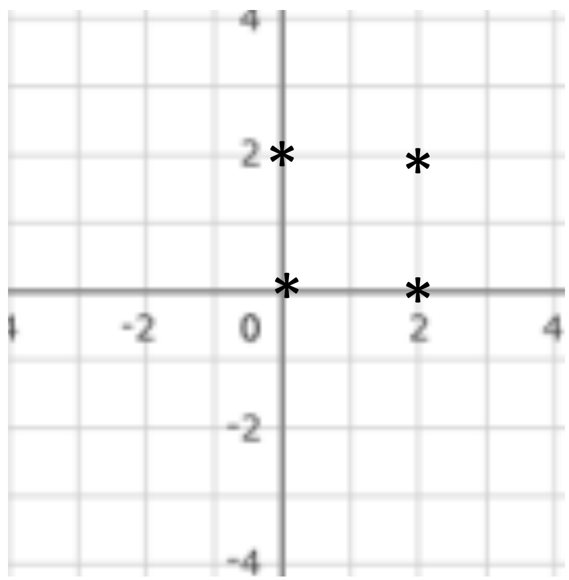
Test Your Understanding

Covariance

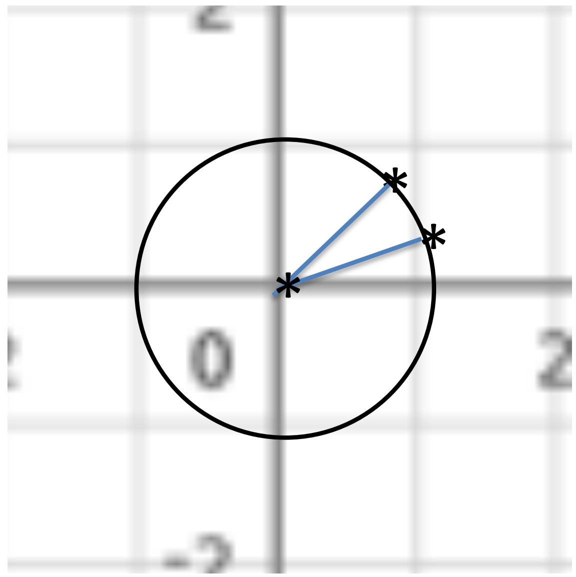
- What is the COV (X,Y) for these points:



(0,0) and (2,2)



(0,0),(2,2), (2,0), (0,2)



Points on a unit circle that are

- 30 degrees and 45 degrees, and the origin
- Or, Every point on the circle

Data X

Multiple inputs and outputs

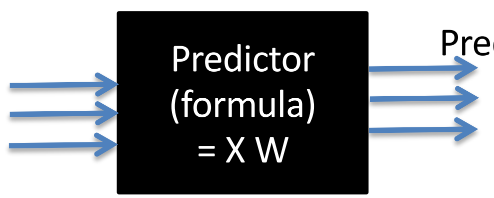
If $Y_{predicted} = f(X,W) = X\ W$

And W is a 4 x 3 matrix, then

a) how many input features (d) are in X?

a) how many outputs (m) are in Y?

X=[x1 x2 ..xd]



y= =[y1 y2 ..ym]

End of Section