Lecture 2:

Vectors + Matrices

We ultimately need to use training data to learn the "best" weight vector Ie, we want $\hat{y}_i = \langle \underline{w}, \underline{x}_i \rangle \approx y_i$ for all $i = 1, \ldots, n$

Our loss function will measure how far each y_i is from $\hat{y}_i$

We can write this objective more simply Define

Label vector $\underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$ and feature matrix or design matrix $X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \in \mathbb{R}^{n \times p}$

real matrix w/ n rows, p columns

$X_{ij} = j^{th}$ feature of i^{th} sample

i^{th} row of X - p features of i^{th} sample $= \underline{x}_i^T$

j^{th} col of X - feature j for all n samples

Then we can write our model as

$$\hat{\underline{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = X\underline{w}$$

$\Leftarrow$ linear model for all n samples in one equation

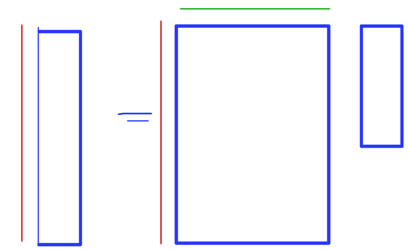
Computing $X\underline{w}$ means taking the inner product of each row of X with $\underline{w}$ and storing the results in a vector $\hat{\underline{y}}$

Note that dimensions should always match

$$\hat{y} = X\underline{w}, \quad \hat{y} \in \mathbb{R}^n, \quad \underline{w} \in \mathbb{R}^p, \quad X \in \mathbb{R}^{n \times p}$$

- # rows of X = length of $\hat{y}$

cols of X = length of $\underline{w}$



Example

$$X = \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 0 & 3 \end{bmatrix}$$

2 features
3 training samples

$$\underline{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \underline{x}_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad \underline{x}_3 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

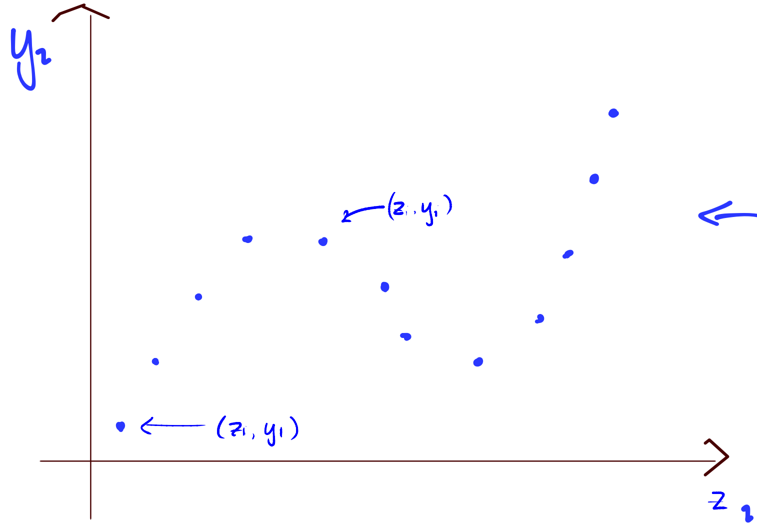
$$\underline{w} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

$$X\underline{w} = \begin{bmatrix} 2\ 1+4\ 0 \\ 2\ 2+4\ 0 \\ 2\ 0+4\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix}$$

Another perspective $X\underline{w}$ is a weighted sum of the columns of X, where $\underline{w}$ gives the weights

$$X\underline{w} = 2\begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} + 4\begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 12 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix}$$

Example



this doesn't look like a straight line, but linear models can still help!

Let $X = \begin{bmatrix} 1 & z_1 & z_1^2 & z_1^3 \\ 1 & z_2 & z_2^2 & z_2^3 \\ \vdots & & \ddots & \vdots \\ 1 & z_n & z_n^2 & z_n^3 \end{bmatrix} \Rightarrow \hat{y} = X\underline{w}$ implies

$$\hat{y}_1 = w_1 \, 1 + w_2 \, z_1 + w_3 \, z_1^2 + w_4 \, z_1^3$$

= cubic polynomial that fits training samples perfectly!

matrices with this special structure are called Vandermonde matrices

Observe (z_i, y_i) for $i = 1, \dots, n$, where z_i, y_i are both scalars

make feature vector $\underline{x}_i = \begin{bmatrix} 1 \\ z_i \\ z_i^2 \\ z_i^3 \end{bmatrix}$

We've looked at multiplying matrices by vectors, but to find good $\underline{w}$ that fit training data, we'll also need to be able to multiply two matrices together. Matrix products are also interesting in their own right.

Example Recommender system

	Becca	Michael	Bo	Victor
Star Wars	6	4	7	5
Pride + Prejudice	4	8	2	6
Oppenheimer	6	2	8	4
Barbie	5	7	4	6
Exorcist	5	3	6	4

$= X$

Let's write X as the product of two matrices U and V

$$\boxed{X} = \boxed{U}\,\boxed{V}$$

$X \in \mathbb{R}^{n \times p}$ $\quad$ $U \in \mathbb{R}^{n \times r}$ $\quad$ $V \in \mathbb{R}^{r \times p}$

Think of

U = taste profiles of r representative customers

V = weights on each representative profile (1 set of weights for each customer)

Example

how much Becca resembles each rep. movie watcher

$$U = \begin{bmatrix} 8 & 3 \\ 0 & 10 \\ 10 & 0 \\ \boxed{3 \quad 8} \\ 7 & 2 \end{bmatrix}, \quad \underline{v} = \begin{bmatrix} 6/10 \\ 4/10 \end{bmatrix} \implies U\underline{v} = \begin{bmatrix} 6 \\ 4 \\ 6 \\ 5 \\ 5 \end{bmatrix}$$

how much action movie lover likes each show

how much romance movie lover likes each show

ratings of Barbie for each representative movie watcher

expected preferences of customer who is 60% action lover and 40% romance lover

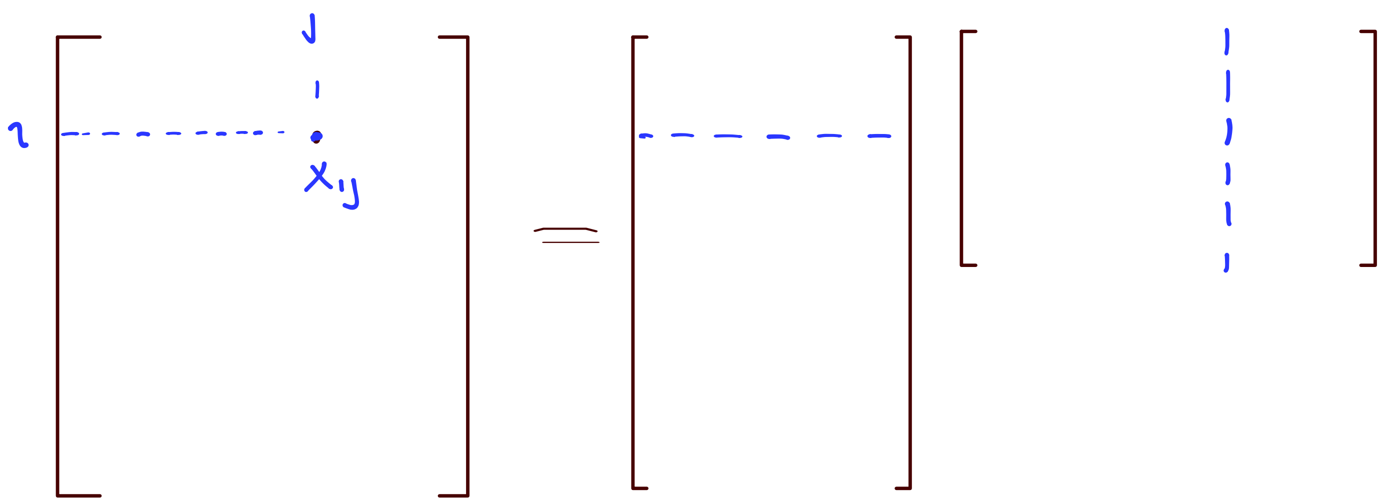
Matrix V will contain a weight vector column for each customer

Example:

$$UV = \begin{bmatrix} 8 & 3 \\ 0 & 10 \\ 10 & 0 \\ 3 & 8 \\ 7 & 2 \end{bmatrix} \begin{bmatrix} 6 & 2 & 8 & 4 \\ 4 & 8 & 2 & 6 \end{bmatrix} \cdot \frac{1}{10}$$

= product of two matrices

If $X = UV$, then $X_{ij} = \langle i^{th}$ row of U, j^{th} col of $V \rangle$



- What is a column of X?

x_j = j^{th} col of X = weighted sum of columns of U, where j^{th} column of V tells us the weights

= $U \underline{v}_j$ = expected tastes of j^{th} customer

- What is a row of X?

x_i = i^{th} row of X = $\underline{u}_i^T V$ = weighted sum of rows of V, where i^{th} row of U tells us the weights

= how much we expect each customer to like i^{th} show

Inner product representation

$$UV = \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix} \begin{bmatrix} | & | & & | \\ V_1 & V_2 & & V_p \\ | & | & & | \end{bmatrix} = \begin{bmatrix} u_1^T V_1 & u_1^T V_2 & & u_1^T V_p \\ & & & \\ u_n^T V_1 & u_n^T V_2 & & u_n^T V_p \end{bmatrix}$$

Outer product representation

$$(UV)_{i,j} = \sum_{k=1}^{r} u_{ik} v_{kj}$$

$$UV = \begin{bmatrix} | & | & & | \\ U_1 & U_2 & & U_r \\ | & | & & | \end{bmatrix} \begin{bmatrix} - & V_1^T & - \\ - & V_2^T & - \\ & \vdots & \\ - & V_r^T & - \end{bmatrix}$$

upper-case U_i = i^{th} column of U

lower-case u_i = i^{th} row of U

$$= U_1 V_1^T + U_2 V_2^T + \cdots + U_r V_r^T$$

for each customer, how much they resemble action lover

ratings for action lover

matrix of action ratings

Given a matrix $X \in \mathbb{R}^{n \times p}$. what is the smallest r such that we can find $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{r \times p}$?

Example : Let $X = \begin{bmatrix} 1 & 3 & 4 \\ 2 & 6 & 8 \\ 3 & 9 & 12 \\ 4 & 12 & 16 \end{bmatrix}$

We could write $X = UV$ with $U = \begin{bmatrix} 1 & 3 & 4 \\ 2 & 6 & 8 \\ 3 & 9 & 12 \\ 4 & 12 & 16 \end{bmatrix}$ and $V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$\Longrightarrow r = 3$ works.

can we find U, V with smaller r ?

consider: $U = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ and $V = \begin{bmatrix} 1 & 3 & 4 \end{bmatrix}$

$\Longrightarrow$ smallest $r = \underline{1}$!

This smallest value of r is the RANK of the matrix X

In the context of our recommeder system example, r is the minimum number of representatative taste profiles we need to accurately represent everyone's movie ratings.