UNIT 2 : Non Linear Classification, Linear Regression and Colaborative Filtering

LECTURE 7 : Recommender Systems

Wooooooooo !!!!!!! We are back baby !!!!! Yeahhhhhh !!!!!

After the successful completion of Fundamentals of Statistics , your pal **Scooby** is back with a brand new set of handwritten lecture notes. I found that many of you liked them and appriciated the work by thanking me personally on my email. Which by the way is :

Scooby's email : **peepeepoopoo873@gmail.com**

Feel free to point out mistakes and message me anyways

For those who do not message me...
I don't care man!!! Whataever man !!!! Screw you guys !!!! Whatever!!!

(I do care, please message me. I was just pretending to be indifferent in the above line so that you can't hurt me emotionally.) Anyways.

I tried to make these notes without any mistakes and cover most of the video material.

But as Yoda has rightly said : ' *Is human, To err* '

I however **DO NOT TAKE** any responsibilty if you fail your exams following my notes. So please check the actual lecture slides and videos if you suspect something is wrong, and mail me too.

The notes **DO NOT CONTAIN** solutions to any homework problems or in-lecture exercises. Only the problems and exercises solved in the videos are included in these notes.

Here is an interesting photo I googled

CUTE BABIES

I ran out of made up quotes so I will write very short stories instead. These may or may not resemble true events, but will definitely have a message that will help you along the journey, that is life.

STORY TIME : Brothers n Blood

Once upon a time a there was a bird that lived in an Amazon rainforest. Her husband who was also a bird(suprise, suprise), had died a long time ago. She was a single mother to three babies.

Ron, Ricky and Rico.
Ron was the studious one. He was the topper of his class.
Ricky was the middle brother. He was very fond of Ron.
Rico was the youngest,one. His parent always pressured him to be more like Ron. He wasn't very bright so he did not perform good in school. He started growing jealous of Ron.

Rico got involved with some shady business in the forest. The people he used to hang out with used to extract an intoxicant from the bark of some tree, and sell it to earn jungle money. They also made contacts with the leopard mafia of the jungle. These were the people who provided them with protection
. Rico was untouchable.

One day Rico came home all wasted, and started shouting at his mother. Ron came out of his room to see what was hapenning. Rico had grabbed her mother by her wing and was forcing her to give him some jungle money in order to buy more jungle drugs. Ron inturrupted in between and slapped Rico right there. Rico took out a gun from his feather pocket and .... BAMMMM!!!... shot Ron right in between the head.
Rico ran away as his mother and Ricky cried on, while the dead body of Ron gushed blood all over their nest. The mother was unable to bear the trauma of her eldest son dying and soon after about a month she died too. The jungle police was unable to arrest Rico due to his connections to leopard mafia.

Ricky saw all this, the death of his mother, his brother and the helplesness of police to bring justice. He cried and cried and when all the tears dried up, he swore to bring justice to his dead mother and dear brother. He went on to train in the inner jungle where a famous shaoilin monkey trained him. He was trained for three years straight to be a deadly killer
After three years he returned for his vengence. He laced the leopard's meat with deadly snake poison (as taught in assasin school). They dead. Then he one by one murdered all the friends of Rico, some were trapped under rocks, some were burnt alive and some were poisoned.

Finally he came to Rico and stood in front of him. He declared to him that he was there to kill him. Rico reached for the gun in his feather pocket, but it was gone. 'Looking for this?' he said, as Ricky pointed Rico's own gun to his head. The same gun that took his elder brothers life. I will count to three he said as Rico stood there paralysed with fear.

One.....Two.....Three.....ZZZZIPPPPPPPP.....ZZZZIPPPPPPPPPP

Came two arrows flying, one to Rico'chest and the other to Ricky's head. Below were the hunteres from a human tribe that lived in the forest.

Ricky and Rico were roasted that night and enjoyed in a tasty meal.

THE END
MORAL : No matter how interesting your story is, if you are tasty, we will eat you
~ sincerely Humans
I hope these notes will help you

Let's get started

▷ Netflix Challenge

In october of 2006, Netflix issued a machine learning challenge with an award prize of 1 million dollars. The problem was in the field of recommender systems, the topic of study for this Lecture. Lets motivate our study by tackling this problem. Lets first understand the problem

Netflix provided a training dataset of the ratings of different movies.
There were around 18,000 movies in this set that were rated.
The ratings were given from 1 to 5 stars (ie integers)

The training examples were of form <user, movie, date of grade, grade>
ie the grade given by that particular user to that movie on date of grade
grade means rating here.

In the data there we ratings from around 500,000 users.

Appendix A

Now come the tricky part

It should be obvious that not every user rated each and every of the 18,000 movies. That's not possible. But Netflix made sure that the train data was such that the average user rated over 200 movies

As not every user will rate every movie, some movies will have many ratings and some will have few ratings. Netflix made sure that on average a movie was rated by over 5000 users.

But we are talking about averages here. There was variance in data
Some movies in the dataset had as few as 3 ratings.
A user rated over 17000 movies

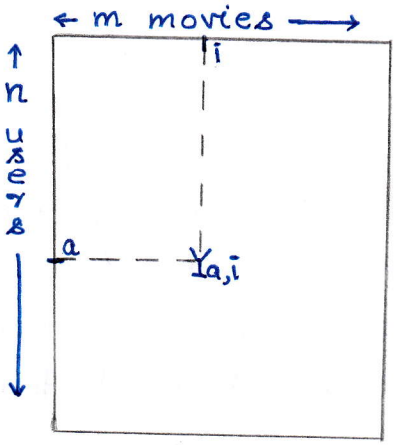
The challenge was to predict the grade/rating of all movies that were part of a test set, whose grades/ratings the jury already knew.

The problem cannot simply be considered a regression problem because:

There are some problems here, let's see

1.st : The train data is not complete only 1.1% of the total ratings were available, because that's how the real world works

2.nd : Let's say we try to mathematically formulate the problem



Suppose I put the ratings in a matrix form
The matrix has m (around 18000) rows for movies
The matrix has n columns (around 500,000) for the users
If we call this matrix Y

$Y_{a,i}$ ≡ Rating given by a^{th} user to i^{th} movie

$Y_{a,i}$ can be NULL (ie no rating)

Now suppose we try to predict $Y_{a,i}$ using regression. If you remember we did classification when we in the start of the first lecture. We said in order to predict whether the ath user will like the ith movie, we will assign a feature vector to each movie like [IMDB rating, budget, cast, Rotten Tomato score ...]^T and will train θ according to the ratings given by ath user. This will give us a regression function that will take in the movie's feature vector as input give a rating for the ath user as output. We can fill the whole matrix using this, by computing each rating for each user for each movie

But there are certain problems with this approach

- In order to predict the ratings for the ath user you need enough ratings by the ath user to train a regression function. Thats not necessarily the case, because ratings are sparse. we don't have all the labels/ratings
- Even if you represent a movie by a feature vector, there are features that can't be encoded properly. For eg. some people might find the ending of Avengers: Endgame happy as everything got back to normal. But some other people may find it sad for some reason (wonder what it is?)
- And the greatest problem of all. That's not the problem statement. Netflix only provides 3 things <user, grade_date, movie> to you as input in the test set. You are not allowed to get different features from different sites for a movie. Netflix won't allow it

This is because the problem is a recommender problem. The idea is about making predictions about the interest of a user by collecting preferences or taste information of many users, that are similiar to the user.

▷ K-Nearest neighbors

So we will now tackle the problem of filling in the missing ratings of test, predicting the rating for a movie for a user with one of the simplest algorithms for instance based learning.

The idea is simple. Suppose we want to predict the rating $Y_{a,i}$, ie the rating user a will give to the ith movie. We will find K people (users) similiar to user a and average there K ratings for the ith movie (under the assumption that all K users have rated ith movie) and we will assign this value as an estimate of $Y_{a,i}$.

How we will do this is a topic for Later.

First lets for the sake of better understanding, lets see an example dataset that I created (all fictional) where I have 8 movies rated by 10 people/users. Not all movies are rated, and the ratings that were not there are marked 0 (in red) as shown in the table. For each of the concept we will need in KNN, we will see alongside how our example relates to it.

	Joker Id : 1	Rocky Id : 2	Inception Id : 3	Gladiator Id : 4	Hangover Id : 5	Watchmen Id : 6	Parasite Id : 7	Sharknado Id : 8
Abigale (a)	3	4	0	4.5	2	0	4.5	5
Borat (b)	4.5	5	3.5	0	3	0	2.5	0
Chloe (c)	0	0	0	4	2	2	4.5	5
Dique (d)	4	0	0	3.5	4.5	2.5	0	0
Emma (e)	4	0	4	0	4	3.5	0	5
Felix (f)	2	5	0	0	0	0	4.5	5
George (g)	3	2	4	5	0	0	0	0
Harry (h)	3.5	1	1.5	3.5	0	3	3.5	0
Ivy (i)	5	4.5	2.5	0	3.5	4.5	0	5
Jameelah (j)	5	3	4	5	0	0	3.5	5

Now lets finiliarize ourselves with this data set.

? What is $Y_{h,7}$

A. It's the rating given by the user Harry (h) to the 7th movie, Parasite its value is 3.5

? What is $Y_{d,6}$?

A. It's the rating given by the user Dique (d) (seems like a frechguy) to the 6th movie Watchmen. It's value is 2.5

? What rating did user f give to Gladiator ?

A. f for Felix ; Gladiator id : 4 so we need to find $Y_{f,4}$
we see $Y_{f,4}$ is 0 meaning f did not give a rating to 4th movie.

Ok now let's explore this idea of K-nearest neighbours. We must keep in mind how we formulated the problem. We want to predict $Y_{a,i}$. ie the rating of ith movie for the user a (Abigale above), so as we discussed we will find K nearest neighbours of a. But what does it mean to find the nearest neighbour of a ? We will need some mathematical term for quantifying this concept of two users being near or similiar. For that we need

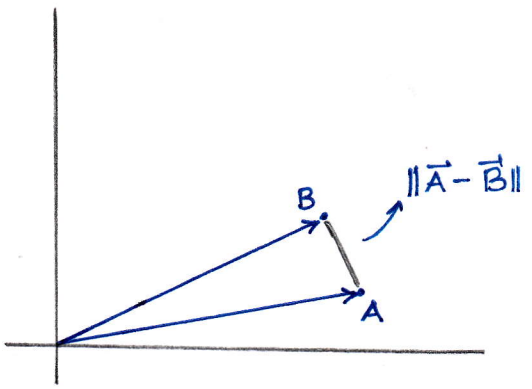
> Similiarity measure

The concept of similiarity measure is a concept that quantifies the similiarity (DUH!!) between two objects. It somehow encapsulates how close or similiar the two objects are. In ML we are intrestested in similarity between two vectors. We will call the first vector x_a and the 2nd vector x_b and we will represent the similiarity between a and b as sim(a,b). We will later see how these vectors are defined according to our movie problem. But first let's see a few similiarity functions

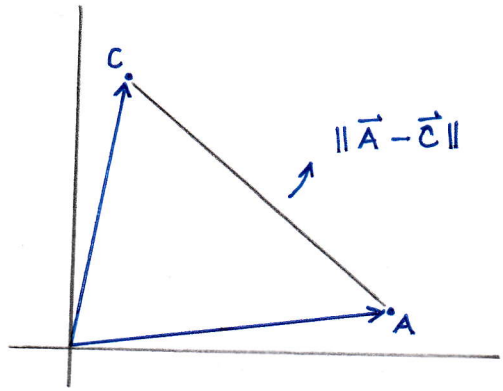
- Related to Euclidean distance

We can find the Euclidean distance b/w the two vectors $\|x_a - x_b\|$ and then form a similiarity measure like $sim(a,b) \propto \frac{1}{\|x_a - x_b\|}$

The farther apart the vectors will be the less will be similiarity measure
The closer the vectors will be, the more similiar will be the vectors



Vector A and B are similiar
They are close so $\|\vec{A}-\vec{B}\|$ ie the distance bw the points is low so $\frac{1}{\|\vec{A}-\vec{B}\|}$ is large



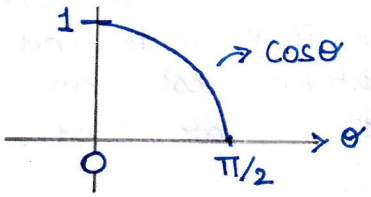
Vectors $\vec{A}$ and $\vec{C}$ are not that similiar
They are far apart so $\|\vec{A}-\vec{C}\|$ ie the distance bw the points is high so $\frac{1}{\|\vec{A}-\vec{C}\|}$ is small

- Cosine Measure

Another similiarity measure that is often used is cosine measure. If the vector x_a and x_b are such that they stay in the 1st quadrant then its not a bad idea to quantify the similiarity bw them. If they point in the same direction, the angle between will be 0, so cosine will be 1.
If they are perpendicular, they will be the most dissimiliar and angle between them will be 90° so cosine ie similiarity will be 0

$$\text{sim}(a,b) = \frac{x_a \cdot x_b}{\|x_a\| \, \|x_b\|}$$ # Cosine Similiarity

This similiarity will go from 0 to 1 as θ goes from 0 to 90°



as vectors come closer and closer

→ vectors come closer →
similiarity bw them increases

Now that we have defined the method/measure to quantify the similiarity between two vectors x_a and x_b lets see how we will define these vectors x_a and x_b in our formulalism

For our problem the vector related to a will be all the ratings : a (Abigale) gave to all of the movies, more precisely

$$x_a = [\,3\,,\,4\,,\,0\,,\,4.5\,,\,2\,,\,0\,,\,4.5\,,\,5\,]^T$$

Lets see again the dataset

	Joker Id : 1	Rocky Id : 2	Inception Id : 3	Gladiator Id : 4	Hangover Id : 5	Watchmen Id : 6	Parasite Id : 7	Sharknado Id : 8
Abigale (a)	3	4	0	4.5	2	0	4.5	5
Borat (b)	4.5	5	3.5	0	3	0	2.5	0
Chloe (c)	0	0	0	4	2	2	4.5	5
Dique (d)	4	0	0	3.5	4.5	2.5	0	0
Emma (e)	4	0	4	0	4	3.5	0	5
Felix (f)	2	5	0	0	0	0	4.5	5
George (g)	3	2	4	5	0	0	0	0
Harry (h)	3.5	1	1.5	3.5	0	3	3.5	0
Ivy (i)	5	4.5	2.5	0	3.5	4.5	0	5
Jameelah (j)	5	3	4	5	0	0	3.5	5

? What is the vector related to user George ie user g.

A. we need to find x_g

$$x_g \equiv [\,3, 2, 4, 5, 0, 0, 0, 0\,]^T$$

Now we are ready to find the nearest neighbours

> Nearest neighbour

Now we are ready to find the nearest neighbours of user a

It's simple we will have to choose a similiarity measure first. Suppose we choose the Euclidean distance. The <u>more</u> the euclidean distance between x_a and x_{other}, the <u>less</u> the similiarity will be between a:user and other user.

Suppose we want to quantify similiarity between user a: Abigale and user e : Emma. we need to find

$$\| x_a - x_e \|$$

$$\left\| \begin{bmatrix} 3 \\ 4 \\ 0 \\ 4.5 \\ 2 \\ 0 \\ 4.5 \\ 5 \end{bmatrix} - \begin{bmatrix} 4 \\ 0 \\ 4 \\ 0 \\ 4 \\ 3.5 \\ 0 \\ 5 \end{bmatrix} \right\| = \left\| \begin{bmatrix} -1 \\ 4 \\ -4 \\ 4.5 \\ -2 \\ -3.5 \\ 4.5 \\ 0 \end{bmatrix} \right\| = \sqrt{1^2 + 4^2 + 4^2 + 4.5^2 + 2^2 + 3.5^2 + 4.5^2 + 0}$$

And if we think about it this similiarity measure is not a bad one

Think about it this way. Suppose some two other users say Adam and Eve rated the movies as follows

$$x_{Adam} = [\,1, 2, 3.5, 0, 4, 5, 4.5, 0\,]$$

$$x_{Eve} = [\,1, 2, 3.5, 0, 4, 5, 4.5\ 0\,]$$

They both have the same vectors, so this means they rated every movie the same that they watched. There were also movies rated 0 ie they did not rate these movies, ie they did not rate the same movies

If x_{Adam} & x_{EVE} is same then $\|x_{Adam} - x_{EVE}\|$ will be 0, ie the Least it can be, ie the similiarity measure will be the most it can ever be. It's like they are the perfect couple, like God made them or something.

So what we will do for our user a : Abigale is calculate this metric $\|x_a - x_j\|$ where $j \in \{b, c, d, e, f, g, h, i, j\}$ and we will see which distance is minimum and sort accordingly

So using highly complex computational tools (numpy) I calculated & sorted these distances from a

$\|x_a - x_j\| = 5.12$

$\|x_a - x_f\| = 5.24$

$\|x_a - x_c\| = 5.41$

$\|x_a - x_h\| = 7.18$

$\|x_a - x_b\| = 8.11$

$\|x_a - x_g\| = 8.34$

$\|x_a - x_i\| = 8.57$

$\|x_a - x_d\| = 8.70$

$\|x_a - x_e\| = 9.47$

Inc dist

This means Abigale (a) and are the most similiar, because they have the minimum distance

And that's no suprise because Abigale & Jameelah had been in an on again off again relationship for the past 3 years.

Abigale keeps on breaking up with Jameelah because he's too reckless and selfish. But a part of her likes that. She somehow feels that she dosen't deserve happiness in life.

Jameelah on the other hand is extremely afraid of being v nerable. Whenever Abigale tries to get emotionally close to him, he percieves it as an attempt by her to take advantange, so he behaves rudely.

OH these two. I tell you..

(I know this cause I made the dataset)

Now we can find any number of nearest neighbours to a

Suppose we want to find the 3 nearest neighbours to a

Those are j, f & c

j → Jame ah

f → Felix

c → Chloe

We can then represent the K nearest neighbours of a as

KNN (a)

So now we are ready to predict Y_{ai}

In our initial approach we will take the K nearest neighbours of a who have rated ith movie and take there average rating as as the predicted estimate

$$\hat{Y}_{ai} = \frac{\sum_{j \in KNN(a)} Y_{j,i}}{K}$$

But we may want to weigh the prediction according to the similiarity a has with different users. A user with which a is more similiar should have more say it the predicted rating, so we can modify

$$\hat{Y}_{a,i} = \frac{\sum_{j \in KNN(a)} sim(a,j)\, Y_{j,i}}{\sum_{j \in KNN(a)} sim(a,j)}$$

▷ But we have a problem here

But how good is this similarity sim(a,b) between two vectors for predicting the rating of different vectors. Suppose Prof. R.B. is interested in ML and Gardening and there is some other person interested in ML but not in Gardening.

The fact is there are certain areas where different users may have same interest, but other areas where users's interest may differ. But a user may rate a particular movie based on his/her particular interests, which may differ from ratings of some other user for the same movie, where the users share the same interests.

Another example is of your wife's boyfriend, you and him share the same lover (your wife), but you are a weak beta, whereas he is a chad who rides a bike and plays guitar. So you have some same same interests, but other characteristics are completely different.

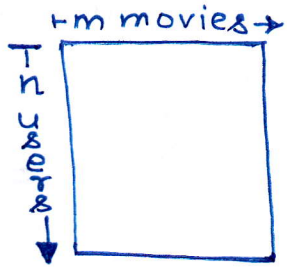
Now let's see a naive approach to tackle this problem via minimization.

▷ Naive Approach

Lets now try to approach this problem as a minimization problem we have the setup as follows

We had our spare matrix Y as before

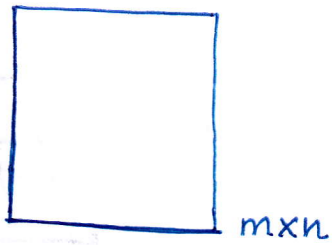
{only few entries are filled in the matrix}



And now we want to make a new matrix, that will contain all our predictions. We will call this matrix X. X will (obviously) be the shape of Y ie a $m \times n$ matrix. and X_{ai} will be the predicted prediction of the ith movie by ath user.

So somehow we will try to minimise something that will give us the entries of X and that something is

X :



Empirical Risk : $J(X)$

We would make this function in the same spirit as earlier risks. There will be a term that will give us the average loss when comparing the predicted values to the real values. But as we know that this can be done for some of the entries of Y because others are empty so we can define a set

$$\mathcal{D} = \{ (a,i) : Y_{a,i} \text{ is given} \}$$

And then we can define the square loss for these cases

$$\sum_{(a,i)\in D} (Y_{ai} - X_{ai})^2/2$$

But we also know the problem of over fitting, so we will have to introduce a regularization term as follows

$$\frac{\lambda}{2} \sum_{(a,i)} X_{ai}^2$$

So the overall empirical risk becomes

$$J(X) = \sum_{(a,i)\in D} (Y_{a,i} - X_{a,i})^2/2 + \frac{\lambda}{2} \sum X_{a,i}^2$$

Now let me tell you something. How many parameters do we want to estimate here? The answer is mn (ie $m \times n$), because we want to estimate the whole matrix X. So we will take the gradient & put it to 0.
The gradient will have a total of mn derivatives of the form $\frac{\partial J(\theta)}{\partial X_{pq}}$
But here we have to consider two scenarios, the first where X_{ai} belongs to $\mathcal{D}$ and the other where X_{pq} does not belong to $\mathcal{D}$

> Scenario 1 $X_{pq} \in \mathcal{D}$ Then $J(\theta)$ will have the terms

$$\ldots\ldots (Y_{pq} - X_{pq})^2/2 + \frac{\lambda}{2} X_{pq}^2 \ldots\ldots$$

In order to avoid confusion I used p,q insted a,i

The other terms of the summation does not depend on X_{pq} (ie ath users rating for ith movie). (We are concerned with partial der wrt X_{pq})

So $$\frac{\partial}{\partial X_{pq}} J(\theta) = \frac{2}{2}(Y_{pq} - X_{pq})(-1) + \frac{2\lambda}{2} X_{pq}$$

Put this to zero we get

$$-Y_{pq} + X_{pq} + \lambda X_{pq} = 0$$

we get $$\boxed{X_{pq} = \frac{Y_{pq}}{1+\lambda} \quad \text{when } X_{pq} \in \mathcal{D}}$$

> Scenario 2 $X \notin \mathcal{D}$ The $J(\theta)$ will have the terms

$$\ldots\ldots \frac{\lambda}{2} X_{pq}^2 \ldots\ldots$$

So $$\frac{\partial}{\partial X_{pq}} J(\theta) = \frac{2\lambda}{2} X_{pq}$$

Put this to zero we get

$$\boxed{X_{pq} = 0 \quad \text{where } X_{pq} \notin \mathcal{D}}$$

So Let me tell yo what result we got

when we already know the rating then $X_{ai} = \frac{Y_{ai}}{1+\lambda}$
ie $X_{ai} \in \mathcal{D}$

when we don't know the rating then $X_{ai} = 0$
ie $X_{ai} \notin \mathcal{D}$

wow what a great prediction. It essentially says, when you do know the rating Distort it by a factor of $1/(1+\lambda)$. And when you don't know the rating just put it equal to 0. That's horrible.

But why did we get this horrible prediction. The answer lies in the way we formulated the problem. Here are some issues

- We wanted to estimate mn parameters X_{ai} using the data matrix Y. But the data matrix was itself a sparse matrix. Ie it had element count very very small compared to mn to begin with.
- Secondly we never talked the idea of similiarity, that we were so interested in bw users. It was as if the choice of one user for any movie had not effect on the choice of another user & vice versa.
- This idea of independence of choice was reflected in our formulation when we took a total of mn partial derivatives & also assumed any X_{ai} was independent of any X_{bj} when $(a,i) \neq (b,j)$.
  so no partial derivative depended on any other X_{bj}.

So we need a better strategy

ARE

YOU

READY ?

HOTTIE !!!!!!!!!!!

But before that we need

▷ Mathematical Interlude : Rank 1 Matrix

Let me remind you what the rank of a matrix is. The rank of a matrix is the number of independent columns in a matrix. It's also the number of independent rows in a matrix. This is because there's a theorem in linear algebra that proves for a matrix A : Row rank(A) = Col Rank(A)

We are talking about general $r \times s$ matrices, and not necessarily square.

There's also another theorem that says, that if A is a $r \times s$ matrix then

$$\text{Rank}(A) \leq \min(r, s)$$

Which means rank of a matrix can be at max the smaller of the two (no of rows (r), no of columns (s))

Ok now look at the following matrix

1. $$\begin{bmatrix} 1 & 2 & 3 \\ 5 & 10 & 15 \end{bmatrix}$$

So it's a 2x3 matrix ie its rank can be at max $\min(2,3) \equiv 2$

So rank of above matrix ≤ 2

Now if I look at the 2nd row its 5 * 1st row. So out of the two rows only one is independent and other (any other) is the multiple of the independent row. Hence Row rank = 1

Similiarly 2nd column = 2 * 1st column
3rd column = 3 * 1st column } So only 1 column is independent

And that's not a suprise because row rank = col rank for any matrix

The column rank of this matrix is still 1.

This is a rank 1 matrix

& there's something interesting about this matrix

If I take two vectors

$$u = \begin{bmatrix} 1 \\ 5 \end{bmatrix} \quad v = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

and take the outer product

$$uv^T = \begin{bmatrix} 1 \\ 5 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 10 & 15 \end{bmatrix}$$

So we made this rank one matrix using the outer product of two vectors

and the u, v were not uniquely defined

If I took some other u, v as

$$\begin{bmatrix} .1 \\ .5 \end{bmatrix} \begin{bmatrix} 10 & 20 & 30 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 10 & 15 \end{bmatrix}$$

If you look at the u, v closely. We obsere that

> We can make any column of our matrix with a multiple of u

$$\begin{bmatrix} 1 & 2 & 3 \\ 5 & 10 & 15 \end{bmatrix} \quad \text{when } u = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$\begin{array}{ccc} \nearrow & \nearrow & \nwarrow \\ 1u & 2u & 3u \end{array}$$

> We can make any row of our matrix with a multiple of v^T

$$\begin{array}{c} 1v^T \rightarrow \\ 5v^T \rightarrow \end{array} \begin{bmatrix} 1 & 2 & 3 \\ 5 & 10 & 15 \end{bmatrix} \quad \text{when } v^T = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

2. Let me take just another example

$$A = \begin{bmatrix} 4 & 5 & 6 \\ 8 & 10 & 12 \end{bmatrix}$$ is also a rank 1 matrix

so we can write it as uv^T

$$A = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 4 & 5 & 6 \\ 8 & 10 & 12 \end{bmatrix}$$

3. Now let's move a little forward and look at the new matrix I have

$$A = \begin{bmatrix} 5 & 7 & 9 \\ 13 & 20 & 27 \end{bmatrix}$$

first of all note that the 1st row is not a multiple of 2nd row which means the two rows are independent therefore

$$\text{Row rank}(A) = \text{Col Rank}(A) = \text{Rank}(A) = 2 \leq \min(2,3)$$

So this is a rank two matrix

Let me carefully write this matrix as follows

$$\begin{bmatrix} 5 & 7 & 9 \\ 13 & 20 & 27 \end{bmatrix} = \begin{bmatrix} 1+4 & 2+5 & 3+6 \\ 5+8 & 10+10 & 15+12 \end{bmatrix}$$

which is equivalent to adding two matrices

$$= \underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 5 & 10 & 15 \end{bmatrix} + \begin{bmatrix} 4 & 5 & 6 \\ 8 & 10 & 12 \end{bmatrix}}$$

These are the same two matrices which we saw earlier in Eq 1 and 2

So we can write this as

$$\begin{bmatrix} 5 & 7 & 9 \\ 13 & 20 & 27 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 \\ 5 \end{bmatrix}}_{\text{Call this } U_1} \underbrace{\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}}_{\text{Call this } V_1^T} + \underbrace{\begin{bmatrix} 1 \\ 2 \end{bmatrix}}_{\text{Call this } U_2} \underbrace{\begin{bmatrix} 4 & 5 & 6 \end{bmatrix}}_{\text{Call this } V_2^T}$$

$$A = U_1 V_1^T + U_2 V_2^T$$

Now let me tell you what I did. We had a rank 2 matrix A and looking carelfully at it we decomposed it as a sum of two rank 1 matrices namely $U_1 V_1^T$ and $U_2 V_2^T$. Can did I know, the way to decompose it such that it comes out as a sum the previous two rank 1 matrices. Well that's because I am a genius and you are not. So shut up. Some may claim that I made the 3rd matrix by adding the two matrices and then working that out backwards but they are wrong. Don't believe them!

Anyways I can write A in a much more neater form

$$A = \begin{bmatrix} 5 & 7 & 9 \\ 13 & 20 & 27 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 5 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

I have just put the u_1 & u_2 in a matrix & the v_1^T & v_2^T in a matrix } then multiplied $\equiv \begin{bmatrix} | & | \\ U_1 & U_2 \\ | & | \end{bmatrix} \begin{bmatrix} - & V_1^T & - \\ - & V_2^T & - \end{bmatrix}$

4. And let we just do 1 more example before I move on to the main point

We have the matrix $A_{4\times5} = \begin{bmatrix} 2 & 3 & 2 & 2 & 1 \\ 1 & 3 & 3 & 2 & 1 \\ 1 & 2 & 3 & 3 & 1 \\ 1 & 2 & 2 & 3 & 2 \end{bmatrix}_{4\times5}$

This is a rank 4 matrix. How do I know this. Simple: Mathematica®
You can check it out yourself. But I don't suggest you do that. Cause that would mean you don't trust me. And I don't know if I can continue this relationship with you if you don't trust me. So anyways, the above Rank 4 matrix A can be broken into 4 rank 1 matrices as follows

$$A = U_1V_1^T + U_2V_2^T + U_3V_3^T + U_4V_4^T$$

$$A = \begin{bmatrix} 2 & 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix}$$

You can again check the 4 matrices above are rank 1 each, but someone have to sleep in the couch tonight. We can write it as

$$A = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 2 \\ 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

i.e

$$A = U_1V_1^T + U_2V_2^T + U_3V_3^T + U_4V_4^T$$

& in a neater form we can write

$$A = \begin{bmatrix} | & | & | & | \\ U_1 & U_2 & U_3 & U_4 \\ | & | & | & | \end{bmatrix}\begin{bmatrix} — V_1^T — \\ — V_2^T — \\ — V_3^T — \\ — V_4^T — \end{bmatrix}$$

Lets call these matrices with $U_1\ U_2\ U_3\ U_4$ as U and with $V_1^T, V_2^T, V_3^T, V_4^T$ as V^T

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

Then

$$A = UV^T$$

So hers the point finally.

Here are the final points

> A rank 1 matrix call it $M_{a \times b}$ can be decomposed as an outerproduct

$$M = U V^T$$

where U is a $a \times 1$ vector
V is a $b \times 1$ vector

> A general matrix call it $A_{r \times}$

Has row rank(A) = col rank(A) $\leq \min(r, s)$

- Ie the no of independent columns = no. of independent rows
- The rank can at max be minimum of (no of rows, no of columns)

> A general matrix call it A of rank say k can be decomposed into a sum of k matrices of rank 1.

$$A = U_1 V_1^T + U_2 V_2^T + \cdots\cdots + U_k V_k^T$$
$$= \sum_{i=1}^{k} U_i V_i^T$$

We can also write it as more neatly as

$$A = UV^T = \begin{bmatrix} | & | & & | \\ U_1 & U_2 & \cdots & U_k \\ | & | & & | \end{bmatrix} \begin{bmatrix} — & V_1^T & — \\ — & V_2^T & — \\ & \vdots & \\ — & V_k^T & — \end{bmatrix}$$

And this brings us to the end of this mathematical interlude. It's been a memorable journey. I will always cherish these moments. See you next time.

▷ BOOM!!!

Suprised you, didn't I? The Lecture is not over yet. we still have to see why this is useful to us.

Let's get back to our matrix X that we were to predict

X generally had n rows for n users and
m columns for m movies

So X was an $n \times m$ matrix. Usually the number of users (n) will be greater than the number of movies (m). Like in Neflix case we had $n \approx 500{,}000$ & $m = 18{,}000$. So the rank of matrix X

$$\text{Rank}(X) = \text{Row rank}(X) = \text{Col rank}(X) \leq \min(n,m)$$

(usually this is m)

Now instead of the older approach lets make an assumption & proceed

▷ X is Rank 1

What does this mean? Why are we doing this?
We will answer these questions a little Later. But first lets for the sake of some clarity assume the we are in a situation with 5 users, 3 movies ie $n = 5$, $m = 3$. So X will look like

	Joker (1)	Rocky (2)	Inception (3)
Abigale (a)			
Borat (b)			
Chloe (c)			
Dique (d)			
Emma (e)			

5×3

Now what does X being of rank 1 mean. It means that only 1 row is independent out of all the 5 above. And I can choose any 1 1 like. say I choose Mr Dique's row. He sounds Frech, he may have good taste in movies

So lets say m I have the following row for Mr. Dique

	Joker (1)	Rocky (2)	Inception (3)
Dique (d)	4	5	2.5

That because I assumed X is rank 1 so every other row of X will the multiple of Mr Dique's row.

So X will look something like

	Joker (1)	Rocky (2)	Inception (3)
Abigale (a)	4	5	2.5
Borat (b)	1.6	2	1
Chloe (c)	1	1.25	0.625
Dique (d)	4	5	2.5
Emma (e)	2	2.5	1.25

5×3

Lets see why would we wanna do it

→ We lost somethings : It's like Mr Dique's choice decides every other users choice (multiplied by a scalar factor) for example

Rating of Joker by Emma = $\frac{1}{2}$ * Rating of Joker by Mr Dique

Rating of Rocky by Emma = $\frac{1}{2}$ * " " Rocky " " "

Rating of Inception by Emma = $\frac{1}{2}$ * " " Inception " " "

And so on for other users

In fact we already know that we can write the rank 1 matrix X as

$X = U V^T$ where U is a 5×1 vector
V is a 3×1 vector

Every column in X will be a multiple of U
Every row in X will be a multiple of V

In this particular case here we are not considering every X_{ai} to be independent of any other X_{bj}. so we lost that freedom

→ We found something : But unlike the previous case where we had to to find out 5×3 elements for X. Here we need to find 5 elements for U vector and 3 elements for V^T vector ie a total of $5+3 = 8$ elements And viola!! we will have our whole $UV^T \equiv X$ matrix

And this is even true for a general $X_{n \times m}$ matrix. Earlier we needed to find $m \times n$ elements. But if we assume rank 1 for X. we just need to find $m+n$ elements for calculating UV^T

So we are now planning to estimate $m+n$ independent elements using our sparse matrix Y.

So aside from the question of this being a good idea or bad idea. Let's first try to solve for X a general $n \times m$ matrix where we assume

> X has Rank 1

We can then write $X = UV^T$

$$X_{n \times m} = U_{n \times 1} \; V^T_{1 \times m}$$

$$\begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{bmatrix} \begin{bmatrix} V_1 & V_2 & \cdots\cdots & V_m \end{bmatrix}$$

So the $X_{ai} = U_a V_i$

X_{ai}: a,i the element of matrix X

U_a: ath element of vector U

V_i: ith element of vector

Let's try to solve / minimise the Empirical risk now

$J(x)$ was defined earlier as

$$J(x) = \sum_{(a,i) \in \mathcal{D}} (Y_{ai} - X_{ai})^2/2 + \frac{\lambda}{2} \sum_{(a,i)} X_{a,i}^2$$

we decided that $X_{ai} = U_a V_i$, we do a little change

$$J(x) = \sum_{(a,i) \in \mathcal{D}} (Y_{ai} - U_a V_i)^2/2 + \frac{\lambda}{2} \sum_{a=1}^{n} U_a^2 + \frac{\lambda}{2} \sum_{i=1}^{m} V_i^2$$

So $J(X)$ is a function of $m+n$ independent parameters namely

m : $V_1 \; V_2 \; \ldots\ldots \; V_m$

n : $U_1 \; U_2 \; \ldots\ldots\ldots \; U_n$

Also note that we have added the terms $\sum_{a=1}^{n} U_a^2 + \sum_{i=1}^{m} V_i^2$ terms insted of $\sum U_a^2 V_i^2$ because U_a & V_i's are independently controlled parameters (Ie we can choose the simultaneously independent of one another) so we want to minimize each one seperately

but minimizing this $J(X)$ is not as simple as earlier suppose

Suppose we are at an a,i such that Y_{ai} is given (not empty)

we will be concerned with

$$\frac{\partial}{\partial u_a}\left\{\frac{(Y_{ai}-u_a v_i)^2}{2} + \frac{\lambda}{2}u_a^2\right\}$$

$$= \frac{2}{2}(Y_{ai} - u_a v_i)(-v_i) + \frac{2\lambda}{2}u_a$$

$$= -Y_{ai}v_i + u_a v_i^2 + \lambda u_a$$

And putting this to zero we get

$$u_a = \frac{Y_{ai}\, v_i}{\lambda + v_i^2}$$

But we don't know v_i so this is not a closed form solution

The equation after taking all the partial derivative & putting them to zero does not give us a matrix system of linear equation. we get these entangled terms like $u_a v_i$ & have to solve for them

So we resort to numerical methods to solve our problem

▷ Numerical Method

Idea is simple. we will first fix v vector & solve for u vector then we will plug the obtained u vector back to solve for u vector

Prof R.B. gives an example

Lets say we have a case of 2 users and 3 movies

The Y looks like $\begin{bmatrix} 5 & ? & 7 \\ 1 & 2 & ? \end{bmatrix}$

we will assume that our predicted matrix $X_{2\times3}$ is a rank 1 matrix

So $X = UV^T = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}\begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$

Our Empirical Risk is

$$J(X) = \sum_{(a,i)\in\mathcal{D}} (Y_{ai} - u_a v_i)^2/2 + \frac{\lambda}{2}\sum_{a=1} u_a^2 + \frac{\lambda}{2}\sum_{i=1} v_i^2$$

Prof R.B chooses the initialization for v

$$v = \begin{bmatrix} 2 \\ 7 \\ 8 \end{bmatrix} \quad \text{this will give us} \quad X = UV^T = \begin{bmatrix} 2u_1 & 7u_1 & 8u_1 \\ 2u_2 & 7u_2 & 8u_2 \end{bmatrix}$$

Again Y was $\begin{bmatrix} 5 & ? & 7 \\ 1 & 2 & ? \end{bmatrix}$

$$J(X) = \frac{1}{2}\left[(5-2u_1)^2 + (7-8u_1)^2 + (1-2u_2)^2 + (2-7u_2)^2\right] + \frac{\lambda}{2}(u_1^2+u_2^2)$$

$$\frac{\partial J(X)}{\partial u_1} = -2(5-2u_1) - 8(7-8u_1) + \lambda u_1 = 0$$

gives us $-10 + 4u_1 - 56 + 64u_1 + \lambda u_1 = 0$

gives us $68u_1 - 66 + \lambda u_1 = 0$

gives us $\boxed{u_1 = 66/(68+\lambda)}$

$$\frac{\partial J(X)}{\partial u_2} = -2(1-2u_2) - 7(2-7u_2) + \lambda u_2 = 0$$

gives us $-2 + 4u_2 - 14 + 49u_2 + \lambda u_2 = 0$

gives us $53u_2 - 16 + \lambda u_2 = 0$

$$\boxed{u_2 = 16/(53+\lambda)}$$

Now remember that λ is given by us before doing all these calculations. So Prof R.B. sets $\lambda = 1$

$$\boxed{\begin{aligned} u_1 &= 66/69 \\ u_2 &= 16/54 \end{aligned}}$$

Now using the vector $u = \begin{bmatrix} 66/69 \\ 16/54 \end{bmatrix}$ we can find the new vector

$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Here again we have

$X = UV^T$ but now U is fixed

$$\begin{bmatrix} 66/69 \\ 16/54 \end{bmatrix} \begin{bmatrix} V_1 & V_2 & V_3 \end{bmatrix} = \begin{bmatrix} \frac{66}{69}V_1 & \frac{66}{69}V_2 & \frac{66}{69}V_3 \\ \frac{16}{54}V_1 & \frac{16}{54}V_2 & \frac{16}{54}V_3 \end{bmatrix}$$

And with this new X I can find expressions for V_1 V_2 V_3 by minimiz. $J(X)$

And so on until the algorithm converges to fixed vectors u and v.

This will finally give us the predicted X matrix

▷ Coming Back : the final analysis

Lets see our analyse if it's a good idea to go via this formalism

Suppose X is a general $n \times m$ matrix with rank 1 Therefore

$$X = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{bmatrix} \begin{bmatrix} V_1 & V_2 & V_3 & \cdots & V_m \end{bmatrix}$$

$$\begin{array}{l} \\ \text{1st user} \\ \text{2nd user} \\ \text{3rd user} \\ \\ \\ \text{nth user} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{1st mov} & \text{2nd mov} & \text{3rd mov} & & \text{mth movie} \end{array} \\ \begin{bmatrix} U_1V_1 & U_1V_2 & U_1V_3 & \cdots & U_1V_m \\ U_2V_1 & U_2V_2 & U_2V_3 & \cdots & U_2V_m \\ U_3V_1 & U_3V_2 & U_3V_3 & \cdots & U_3V_m \\ \vdots & & & & \\ U_nV_1 & U_nV_2 & U_nV_3 & \cdots & U_nV_m \end{bmatrix} \end{array}$$

We see a pattern here for the ath user, the rating vector looks like

$$\begin{array}{l} \\ a\text{th user} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{1st mov} & \text{2nd mov} & \text{3rd mov} & & \text{mth movie} \end{array} \\ \begin{bmatrix} U_aV_1 & U_aV_2 & U_aV_3 & \cdots & U_aV_m \end{bmatrix} \end{array}$$

This 'u_a' is influencing, or we can say encapsulating the characteristic of user a. Unless the rating is not from user a, u_a is not used anywhere else.

Similiarly if we see all the rating associated to movie i in a column form

$$\begin{array}{l} \\ \text{1st user} \\ \text{2nd user} \\ \text{3rd user} \\ \\ \text{nth user} \end{array} \begin{array}{c} \text{ith movie} \\ \begin{bmatrix} U_1V_i \\ U_2V_i \\ U_3V_i \\ \vdots \\ U_nV_i \end{bmatrix} \end{array}$$

This 'V_i' is encapsulating the characteristic of movie i

Unless the rating is not for movie i, V_i is not used anywhere else.

So this gives us a new point of view for our formalism

each u_a : represents a characteristic specific to the ath user.
each v_i : represents a characteristic specific to the ith movie.

So u vector can be seen as

$$u = \begin{bmatrix} \text{Characteristic of user 1} \\ \text{Characteristic of user 2} \\ \vdots \\ \text{Characteristic of user } n \end{bmatrix}_{n \times 1}$$

Similiarly v vector can be seen as

$$v = \begin{bmatrix} \text{characteristic of 1st movie} \\ \text{characteristic of 2nd movie} \\ \vdots \\ \text{Characteristic of mth movie} \end{bmatrix}_{m \times 1}$$

So in a way we are finding 1 characteristic (independent of other) for each user and similiarly for each movie when we are finding $u_1\ u_2 \ldots u_n$ & $v_1\ v_2 \ldots v_m$

> But what if we want to say rank(X) > 1

we can do that, in fact we did the analysis for rank(X) = 1 just for the sake of simplicity.

Let's see how the problem will evolve in case when Rank (X) = 2

we already know from our discussions on rank one matrices that X can be decomposed as a sum of two rank 1 matrices or equivalently we can write them in a neat algebraic form

$$X = UV^T = \begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix} \begin{bmatrix} — v_1^T — \\ — v_2^T — \end{bmatrix}$$

I will again write X as follows:

$$X = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \\ u_{31} & u_{32} \\ \vdots & \vdots \\ u_{n1} & u_{n2} \end{bmatrix} \begin{bmatrix} v_{11} & v_{21} & v_{31} & \cdots\cdots & v_{m1} \\ v_{12} & v_{22} & v_{32} & \cdots\cdots & v_{m2} \end{bmatrix}$$

$$\begin{array}{c} \\ \text{1st user} \\ \text{2nd user} \\ \text{3rd user} \\ \\ \\ \text{nth user} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{1st movie} & \text{2nd movie} & & & \text{mth movie} \end{array} \\ \begin{bmatrix} u_{11}v_{11} + u_{12}v_{12} & u_{11}v_{21} + u_{12}v_{22} & \cdots & u_{11}v_{m1} + u_{12}v_{m2} \\ u_{21}v_{11} + u_{22}v_{12} & u_{21}v_{21} + u_{22}v_{22} & \cdots & u_{21}v_{m1} + u_{n2}v_{m2} \\ u_{31}v_{11} + u_{32}v_{12} & u_{31}v_{21} + u_{32}v_{22} & \cdots & u_{\ }v_{m1} + u_{32}v_{m2} \\ \vdots & \vdots & \vdots & \vdots \\ u_{n1}v_{11} + u_{n2}v_{12} & u_{n1}v_{21} + u_{n2}v_{22} & \cdots & u_{n1}v_{m1} + u_{n2}v_{m2} \end{bmatrix} \end{array}$$

And again look at the vectors

$$\text{ath user} : \begin{bmatrix} u_{a1}v_{11} + u_{a2}v_{12} & u_{a1}v_{21} + u_{a2}v_{22} & \cdots\cdots & u_{a1}v_{m1} + u_{a2}v_{m2} \end{bmatrix}$$

Here we have u_{a1} & u_{a2} elements that encapsulate <u>two</u> characteristics of user a
They will never appear on any other rating unless its of user a

Similiarly for the ith movie

$$\begin{bmatrix} u_{11}v_{i1} + u_{12}v_{i2} \\ u_{21}v_{i1} + u_{22}v_{i2} \\ u_{31}v_{i1} + u_{32}v_{i2} \\ \vdots \\ u_{n1}v_{i1} + u_{n2}v_{i2} \end{bmatrix}$$

v_{i1} and v_{i2} encapsulate <u>two</u> characteristics of the ith movie.
These two will never appear for any other movie's rating in X

So Just like we did before we can view this formulation by the point of view of U & V as

$$U = \begin{bmatrix} \text{1st Char. of 1st user} & \text{2nd Char of 1st User} \\ \text{1st Char. of 2nd user} & \text{2nd Char of 2nd user} \\ \text{1st Char. of 3rd user} & \text{2nd Char of 3rd user} \\ \vdots & \\ \text{1st Char of nth user} & \text{2nd Char. of nth user} \end{bmatrix}_{n\times 2}$$

$$V = \begin{bmatrix} \text{1st Char. of 1st movie} & \text{2nd Char. of 1st movie} \\ \text{1st Char. of 2nd movie} & \text{2nd Char. of 2nd movie} \\ \text{1st Char. of 3rd movie} & \text{2nd Char. of 3rd movie} \\ \vdots & \\ \text{1st Char. of mth movie} & \text{2nd Char. of mth movie} \end{bmatrix}_{m\times 2}$$

So when we are findind U and V we are finding two independent characteristics for n users each & similiarly two independent characteristics for m movies each.

& Finally let me tell you. It's we who choose the rank(X) and by this way we can choose any K, to encapsulate k characteristics for each user & for each movie we can always then form X as

$$X = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_k \\ | & | & & | \end{bmatrix}_{n\times k} \begin{bmatrix} - & V_1^T & - \\ - & V_2^T & - \\ & \vdots & \\ - & V_k^T & - \end{bmatrix}_{k\times m}$$

& minimise.

$$J(U,V) = \underbrace{\frac{1}{2}\sum_{(a,i)\in D}\left(Y_{ai} - [UV^T]_{ai}\right)^2}_{\text{Squared Error}} + \underbrace{\frac{\lambda}{2}\sum_{a=1}^{n}\sum_{j=1}^{k} U_{aj}^2 + \frac{\lambda}{2}\sum_{i=1}^{m}\sum_{j=1}^{k} V_{ij}^2}_{\text{Regularization}}.$$

This is an extremely powerful method for which K is chosen according to the amount of data we have & number of characterstics we want to incorporate in our formulism.

FIN.

Appendix A Netflix Prize WIKIPEDIA

The **Netflix Prize** was an open competition for the best collaborative filtering algorithm to predict user ratings for films, based on previous ratings without any other information about the users or films, i.e. without the users or the films being identified except by numbers assigned for the contest.

The competition was held by Netflix, an online DVD-rental and video streaming service, and was open to anyone who is neither connected with Netflix (current and former employees, agents, close relatives of Netflix employees, etc.) nor a resident of certain blocked countries (such as Cuba or North Korea).[1] On September 21, 2009, the grand prize of US$1,000,000 was given to the BellKor's Pragmatic Chaos team which bested Netflix's own algorithm for predicting ratings by 10.06%.[2]

Problem and data sets

Netflix provided a *training* data set of 100,480,507 ratings that 480,189 users gave to 17,770 movies. Each training rating is a quadruplet of the form `<user, movie, date of grade, grade>`. The user and movie fields are integer IDs, while grades are from 1 to 5 (integral) stars.[3]

The *qualifying* data set contains over 2,817,131 triplets of the form `<user, movie, date of grade>`, with grades known only to the jury. A participating team's algorithm must predict grades on the entire qualifying set, but they are only informed of the score for half of the data, the *quiz* set of 1,408,342 ratings. The other half is the *test* set of 1,408,789, and performance on this is used by the jury to determine potential prize winners. Only the judges know which ratings are in the quiz set, and which are in the test set—this arrangement is intended to make it difficult to hill climb on the test set. Submitted predictions are scored against the true grades in terms of root mean squared error (RMSE), and the goal is to reduce this error as much as possible. Note that while the actual grades are integers in the range 1 to 5, submitted predictions need not be. Netflix also identified a *probe* subset of 1,408,395 ratings within the *training* data set. The *probe*, *quiz*, and *test* data sets were chosen to have similar statistical properties.

In summary, the data used in the Netflix Prize looks as follows:

- Training set (99,072,112 ratings not including the probe set, 100,480,507 including the probe set)
  - Probe set (1,408,395 ratings)
- Qualifying set (2,817,131 ratings) consisting of:
  - Test set (1,408,789 ratings), used to determine winners
  - Quiz set (1,408,342 ratings), used to calculate leaderboard scores

For each movie, title and year of release are provided in a separate dataset. No information at all is provided about users. In order to protect the privacy of customers, "some of the rating data for some customers in the training and qualifying sets have been deliberately perturbed in one or more of the following ways: deleting ratings; inserting alternative ratings and dates; and modifying rating dates".[2]

The training set is such that the average user rated over 200 movies, and the average movie was rated by over 5000 users. But there is wide variance in the data—some movies in the training set have as few as 3 ratings,[4] while one user rated over 17,000 movies.[5]

There was some controversy as to the choice of RMSE as the defining metric. Would a reduction of the RMSE by 10% really benefit the users? It has been claimed that even as small an improvement as 1% RMSE results in a significant difference in the ranking of the "top-10" most recommended movies for a user.[6]