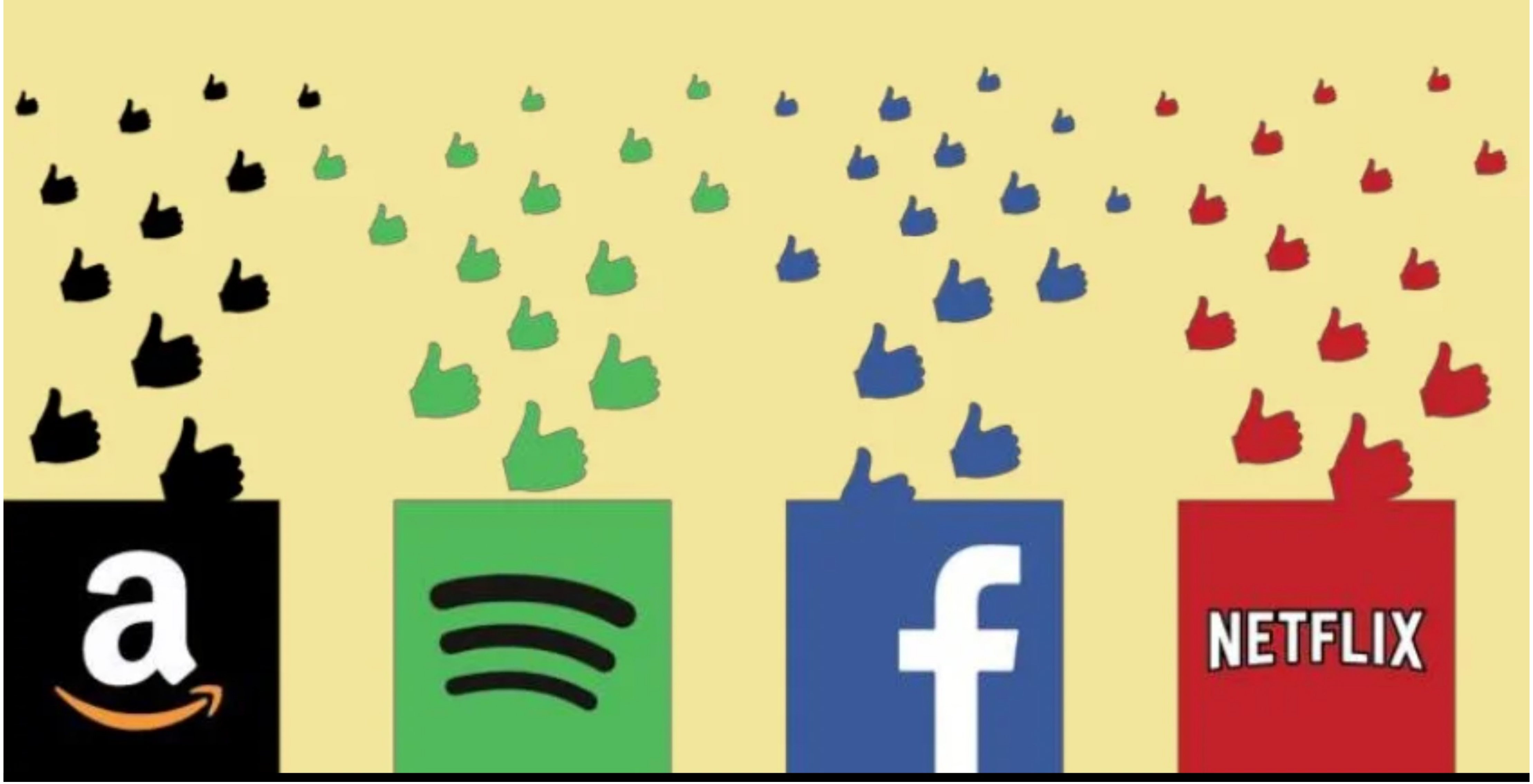

NETFLIX

WHY DO WE USE IT ?

Better Customer Experience

User Personalization

Increase Revenue

Representations

A (typically) low-dimensional vector that encodes the feature information about the user or item.

Often called “embedding,” “latent user/item,”or “latent representation”.

”Representation size, which is the dimension of the latent space, is often referred to as “components.”

Example : Matrix Representation

$$
\begin{array}{cc}
 & \begin{array}{cccccc} \text{Item 1} & \text{Item 2} & \text{Item 3} & \text{Item 4} & \text{Item 5} & \text{Item 6} \end{array} \\
\begin{array}{c} \text{User 1} \\ \text{User 2} \\ \text{User 3} \\ \text{User 4} \end{array} &
\begin{bmatrix}
1.0 & 1.0 & & & -1.0 & 1.0 \\
1.0 & & 0.5 & 1.0 & -0.5 & \\
 & & & & 1.0 & \\
0.75 & & 1.0 & & &
\end{bmatrix}
\end{array}
$$

THREE MAJOR METHODOLOGY

Content Based

Collaborative Filtering

Hybrid Systems

Content Based Recommendation Systems

Filter the key topics from the document where user interested and interacted with it and train the model with those keywords to provide the relevant document to the user.

Example : Content Based

	Action	Rescue	President
User 1	1	1	0
User 2	1	0	1

User Profile

	Action	Rescue	Preseident
Olympus	1	1	1
White House	1	1	0
London	1	0	1

Item Profile

	Olympus	White House	London
User 1	2	2	?
User 2	?	1	2

User Item Interaction

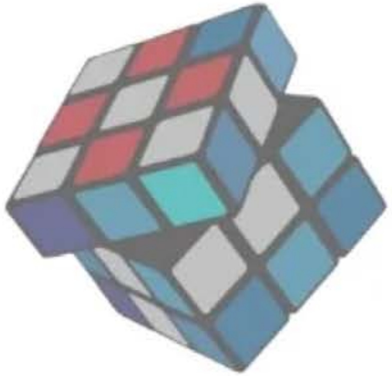
Collaborative Filtering

Identifying the similarity with user and items interactions and find the best similar user/item for the target user , this similarity data act as the interaction dataset for the recommendation systems .

Example : Collaborative Filtering

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5
User 1	2	?	4	3	
User 2	2	2	3		4
User 3	1	2	3	2	

User Item Interactions



Hybrid Recommendation Systems

A system that combines content-based filtering and collaborative filtering could take advantage from both the representation of the content as well as the similarities among users.

Framework of Recommendation Systems

- Interactions
- User Features
- User Representations
- Item Features
- Item Representations
- Prediction
- Learning
- Evaluation

Evaluation Metrics

Precision: The fraction of total no of relevant items is there in the recommended items by the recommender system

Recall: The fraction of total no of relevant items is recommended from the relevant items list by the recommender system

AUC_Score: The probability that a randomly chosen positive example has a higher score than a randomly chosen negative example . A perfect score is 1.0.

LightFM

LightFM is a Python implementation of a number of popular recommendation algorithms for both implicit and explicit feedback, including efficient implementation of **BPR** and **WARP** ranking losses. It's easy to use, fast (via multithreaded model estimation), and produces high quality results.

FRAMEWORK OF LIGHTFM

Interactions	*
User Features	*
User Representation	Linear
Item Features	*
Item Representation	Linear
Prediction	Dot-Product
Learning	Logistic, WARP,BPR

Example :

	Aerospace	Medicine	Analytics	Transport
John	1	-1	1	1
Laura	-1	1	1	0
Tim	0	-1	1	1

User Profile Representation

	Aerospace	Medicine	Analytics	Transport
flight project	1	0	1	1
Drug Discovery	0	1	1	0
Automobile Incubation	0	0	1	1

Item Profile Representation

After taking Dot Product (Making Prediction),

	Flight project	Drug Discovery	Automobile Incubation
John	1	-1	0
Laura	-1	1	0
Tim	0	-1	1

1 = User Liked it
-1 = User Dislike it
0 = Predict the user recommendation level and the user not interacted with it

Learn using Loss Functions

Four Kinds of Loss Function used to optimize the recommendations

- **Logistic**
- **Bayesian Personalized Ranking (BPR)**
- **Weighted Approximate Pair Wise (WARP)**
- **K-OS WARP**

Logistic : Used when both +ve & -ve

Example: Consider you have to give the probability on how much automobile incubation can be recommended to John

Find X: Calculate the dot product of Automobile Features and John Features

i.e : X = 1*0 + -1 *0 + 1*1 + 1*1
= 2

Logistic Function f(x) : $(1 + e^{-x})^{-1}$: $(1 + e^{-2})^{-1}$: **0.73**

WARP (Example)

Consider we are recommending the project for tim

	Flight Project	Drug Discovery	Automobile
Tim	0.2	0.57	0.6
Actual Output	0	1	0

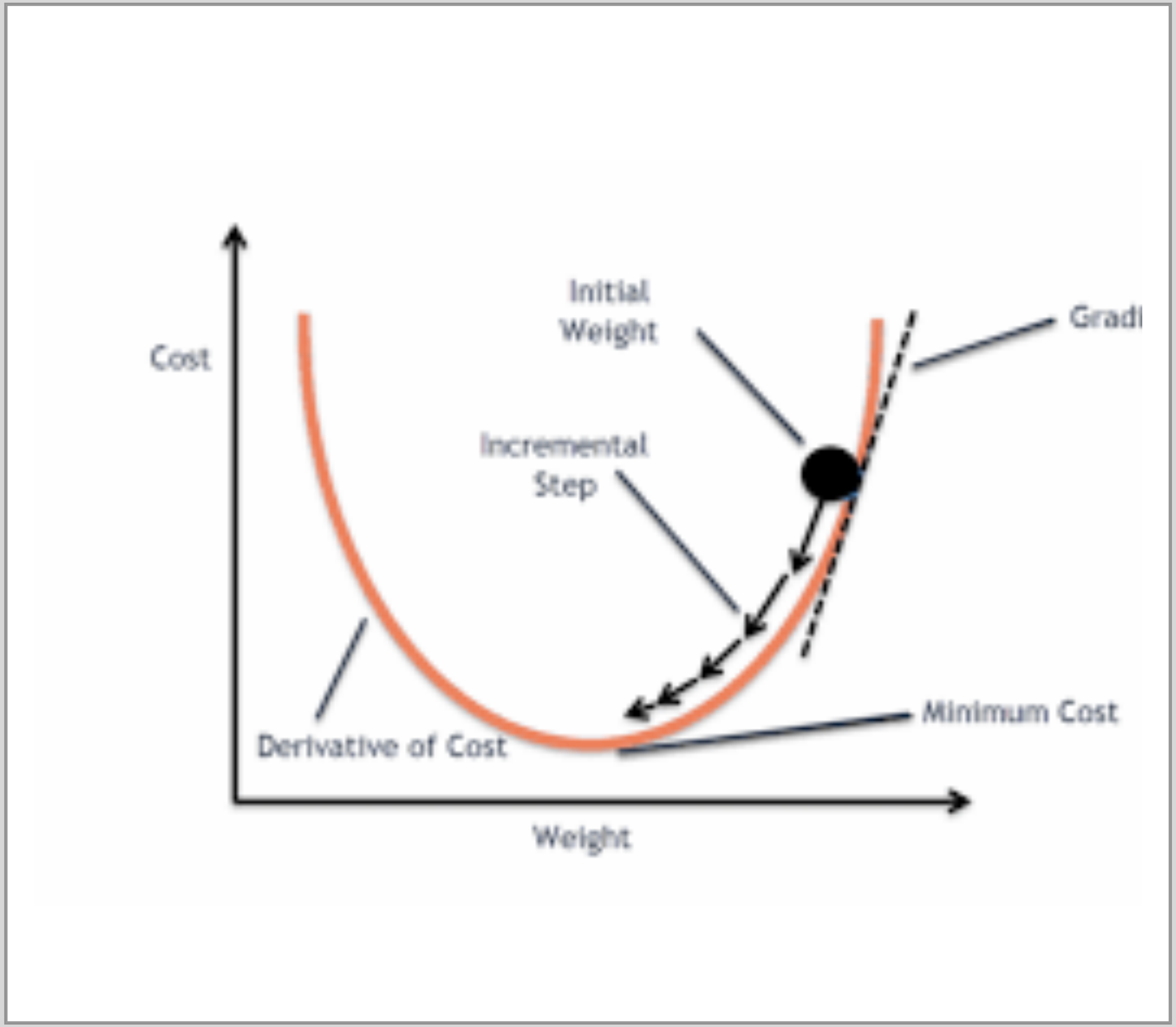
x1: Flight Project x2: Drug Discovery x3: Automobile

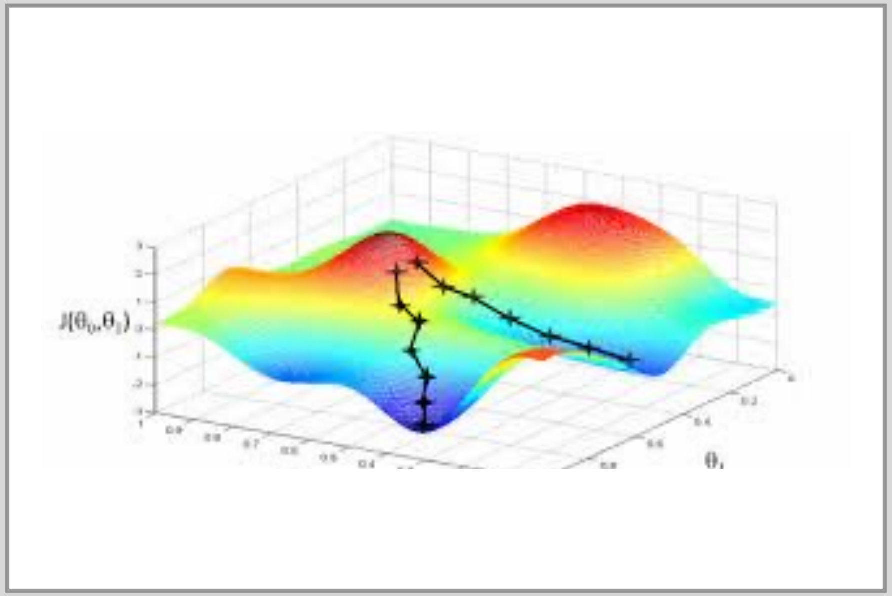
Pairwise(x1,x2) : x1 < x2 : x2 (output) No loss

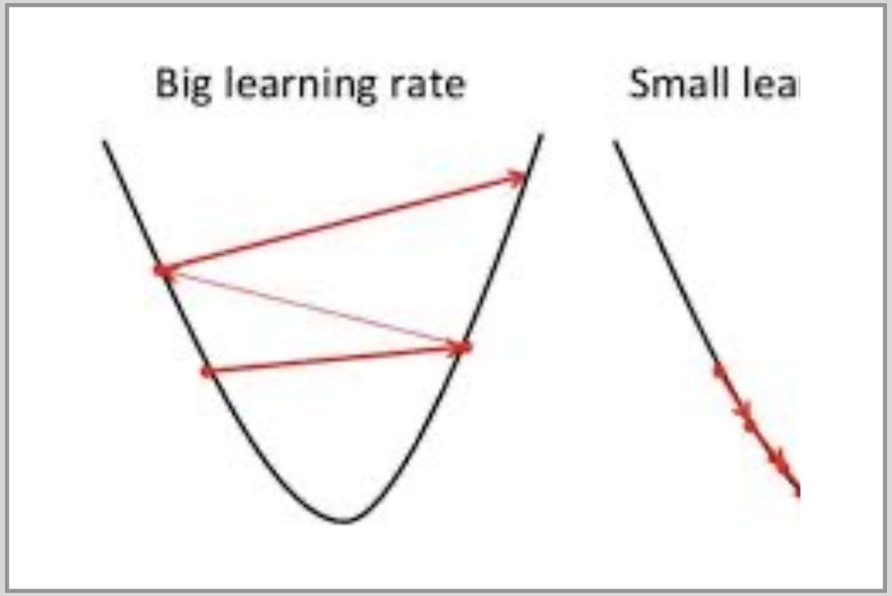
Pairwise(x2,x3) : x3 > x2 : x3 (Output) Loss(x2 Output)

Loss Function (x2,x3)=ln(X-1/N)(x3 - x2) = ln (3 - 1 /2) (0.6 - 0.2) = 0

To optimize the loss use **stochastic gradient**


Initial
Weight
Gradi
Cost
Incremental
Step
Minimum Cost
Derivative of Cost
Weight




Big learning rate
Small lea

Benchmarking of LightFM