Recommending Relevant Products to a User

Swetha Kolalapudi
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Overview

Understand the role of Personalized Recommendations

Predict user-product ratings using Collaborative Filtering

Find hidden factors that influence user-product ratings

Implement Movie Recommendations in Python

Personalized Recommendations

Product recommendations on Amazon

Movie recommendations on Netflix

Gmail auto-tagging important e-mails

Personalization

The new trend across all online services

Inbox organization

Facebook Newsfeed

New York Times homepage

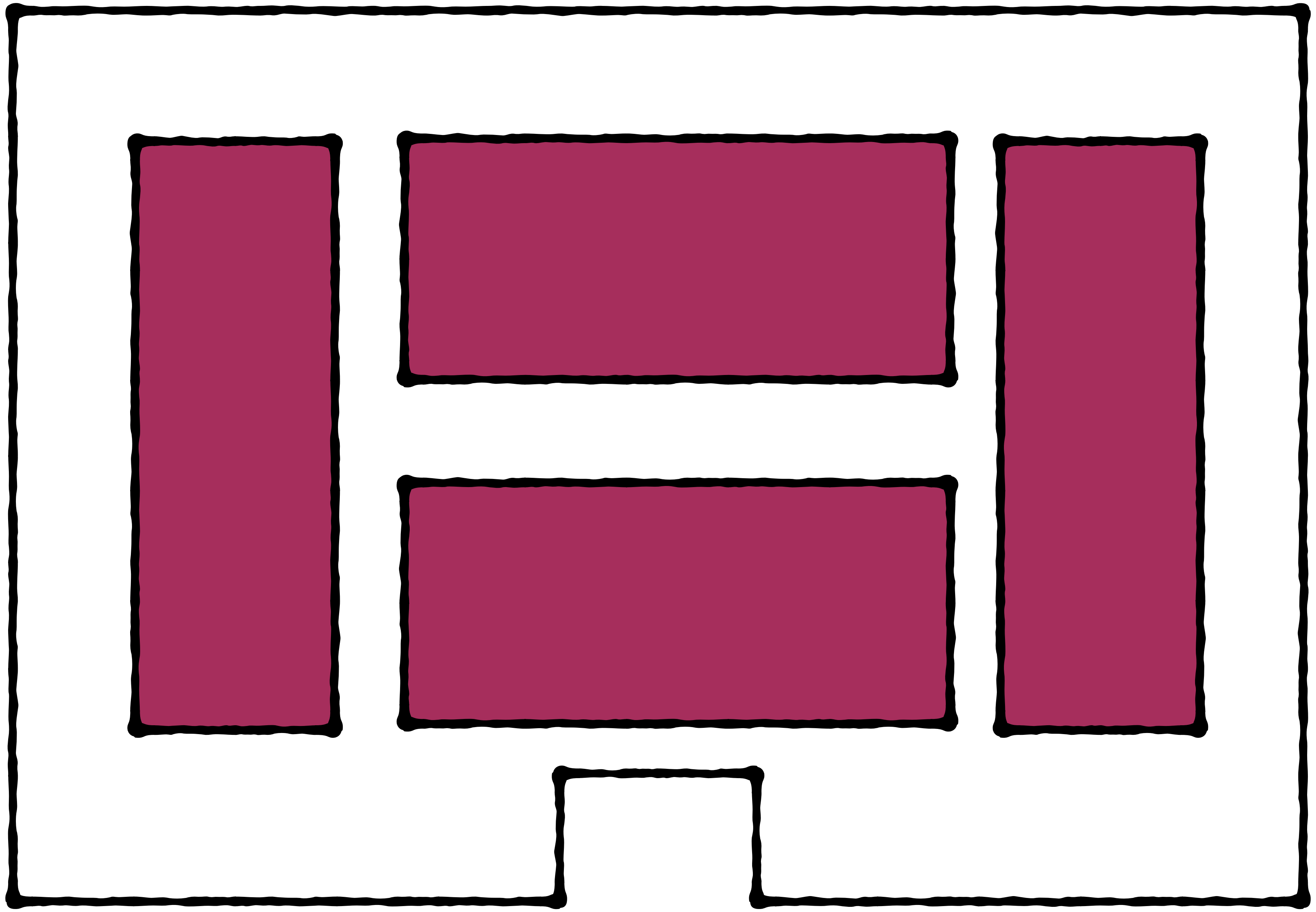
Personalization

Webpages and online services used to be static

..just like an offline store

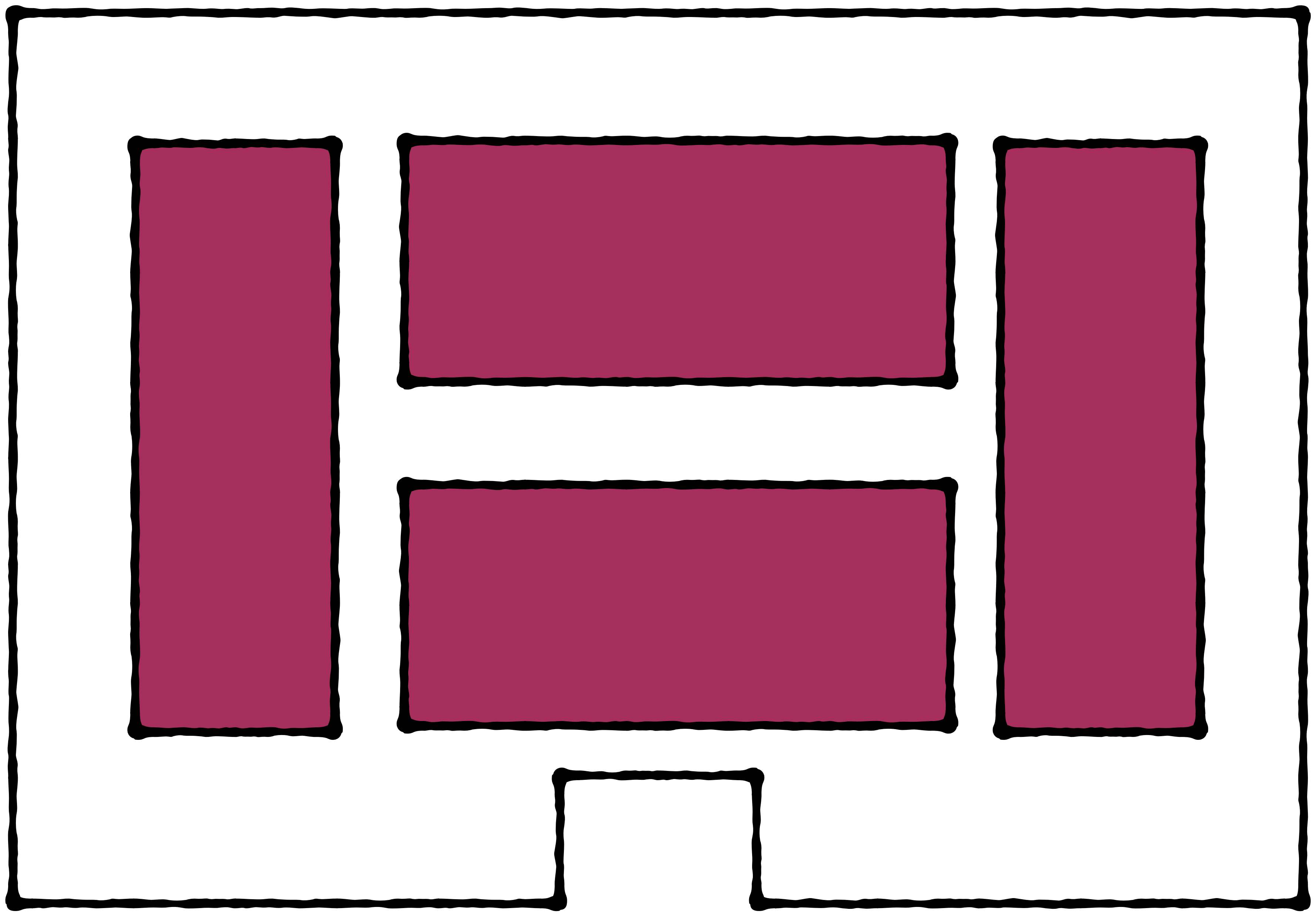
Personalization

Offline stores are designed to appeal to **the majority** of users



Personalization

What if, the store could change, so that each user sees the design that most appeals to them?



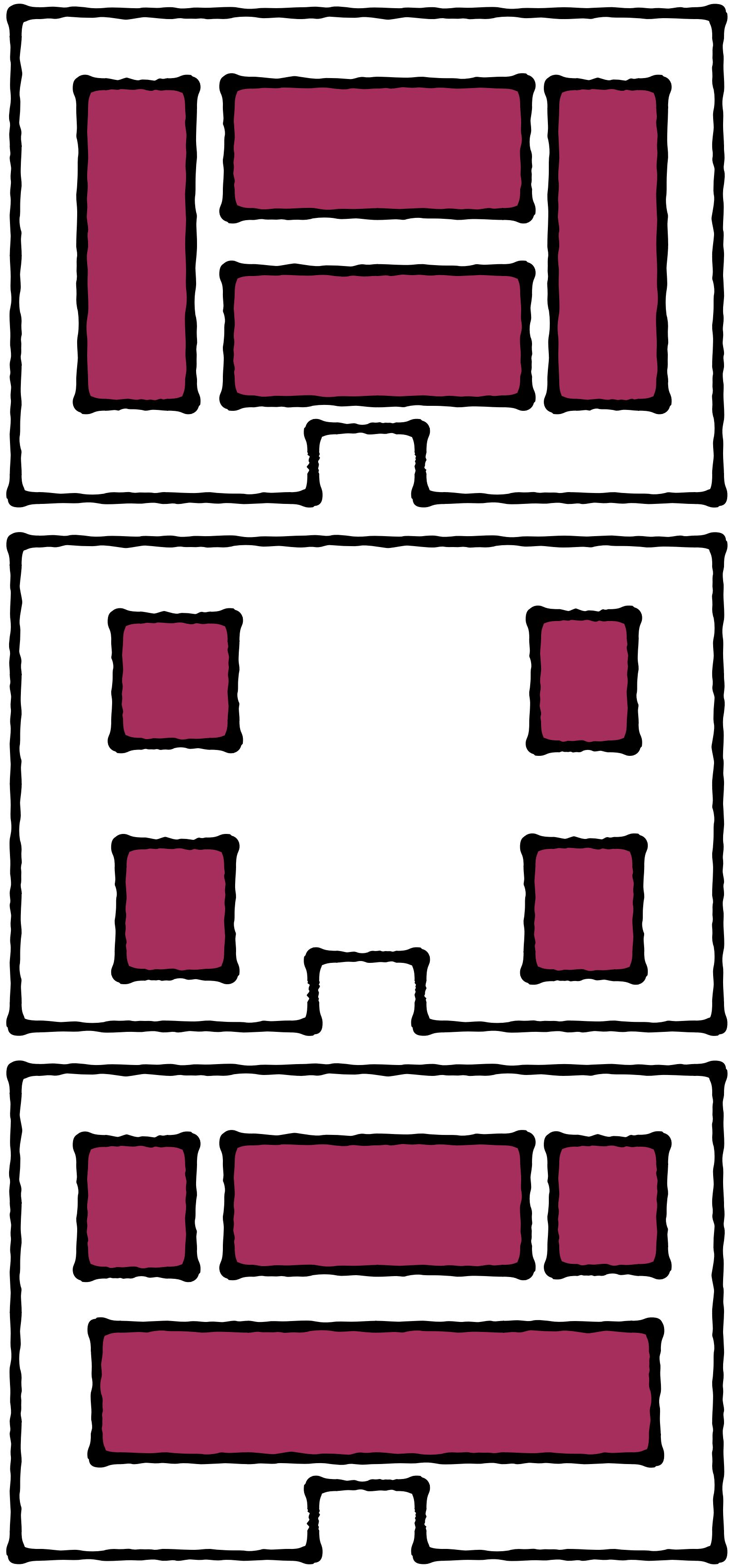
Personalization

What if, the store could change, so that each user sees the design that most appeals to them?

User 1

User 2

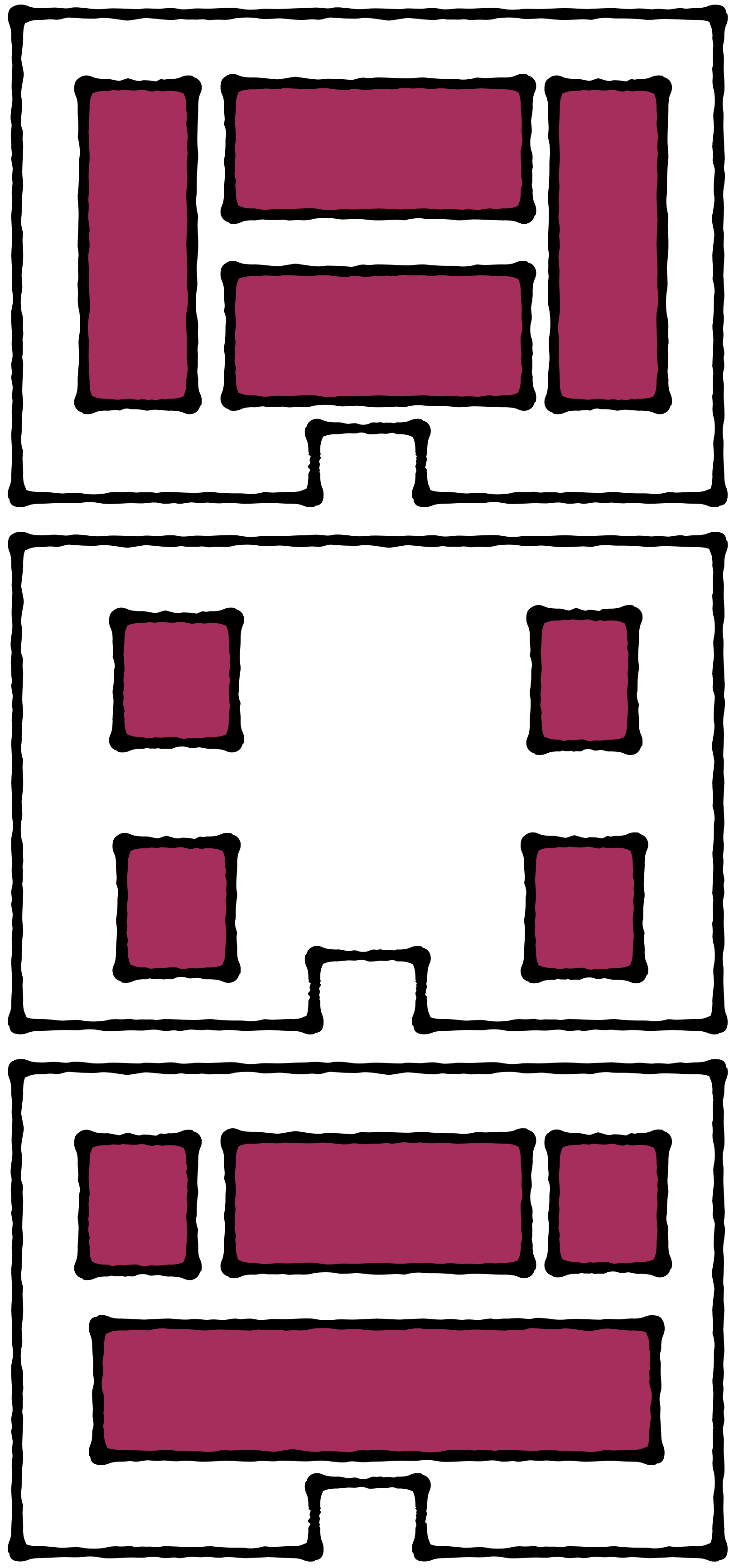
User 3



Personalization

The store is **personalized** based on

1. The user's preferences
2. The user's needs
3. What the user is currently looking for



Personalization

Discovery

Engagement

Discovery

Unlike offline stores, online stores have **huge catalogs**

Millions of Books	Thousands of Songs	Hundreds of News articles	Thousands of Videos

Discovery

Users need help finding
what they are looking for

..sometimes things
they didn't know
they were looking for

Engagement

The more time users spend at a website

..the more likely they are to open their wallets

Example 1: Find Top 10 Movie Picks for a User

1. Compute the ratings for all movies for all users

2. Sort movies for each user in descending order based on their ratings

3. Pick the top 10 movies this user hasn’t watched

Example 2: Recommendations Based on Browsing History

1. Use # views of a product as an implicit rating

2. Compute the ratings for all products for all users

3. Pick the top 10 products for this user

Example 3: Users Who Bought This Also Liked....

1. Compute the ratings for all products for all users

2. Subset the computed ratings to users who have bought this product

3. Pick the 10 products with highest ratings for this subset

The Common Problem to Solve

Compute the ratings for all products for all users

Collaborative Filtering Algorithms

Predict user ratings for products based on a user's past behavior

User purchases

User Browsing history

User clicks

User Ratings, Reviews

Collaborative Filtering

Top picks for you!!

If you like this, you'll love that!!

If you've bought this, you'll need that!

Collaborative Filtering

A general term for any algorithm that only uses past user behavior for identifying recommendations

Collaborative Filtering

The basic premise is that

If 2 users have the same opinion about a bunch of products

They are likely to have the same opinion about other products too!

Collaborative Filtering

The training data should be of the form

User Id	Product Id	Rating

Collaborative Filtering

User Id	Product Id	Rating

Books, Videos, Movies, Artists, News Articles, e-mail keywords

Collaborative Filtering

User Id | Product Id | **Rating**

A measure of a
user's preference for
a particular product

Collaborative Filtering

Explicit Ratings	Collected at the store or through e-mail surveys
Implicit Ratings	#Clicks, #Purchases, #Shares, #Likes, # Times watched

Collaborative Filtering

User Id	Product Id	Rating

The algorithm will compute a rating for all products for every user

Latent Factor Analysis

A subset of Collaborative Filtering algorithms

Identify hidden factors that influence user behavior

Hidden Factors for Movies

Genre	Cast Popularity
Commercial Appeal	Recency of release

Hidden Factors for Movies

Genre

Commercial Appeal

Cast Popularity

Recency of release

1. Rate every User on their preference for these factors

2. Rate every Movie on a scale of 1 to 5 for these factors

Hidden Factors for Movies

Genre

Commercial Appeal

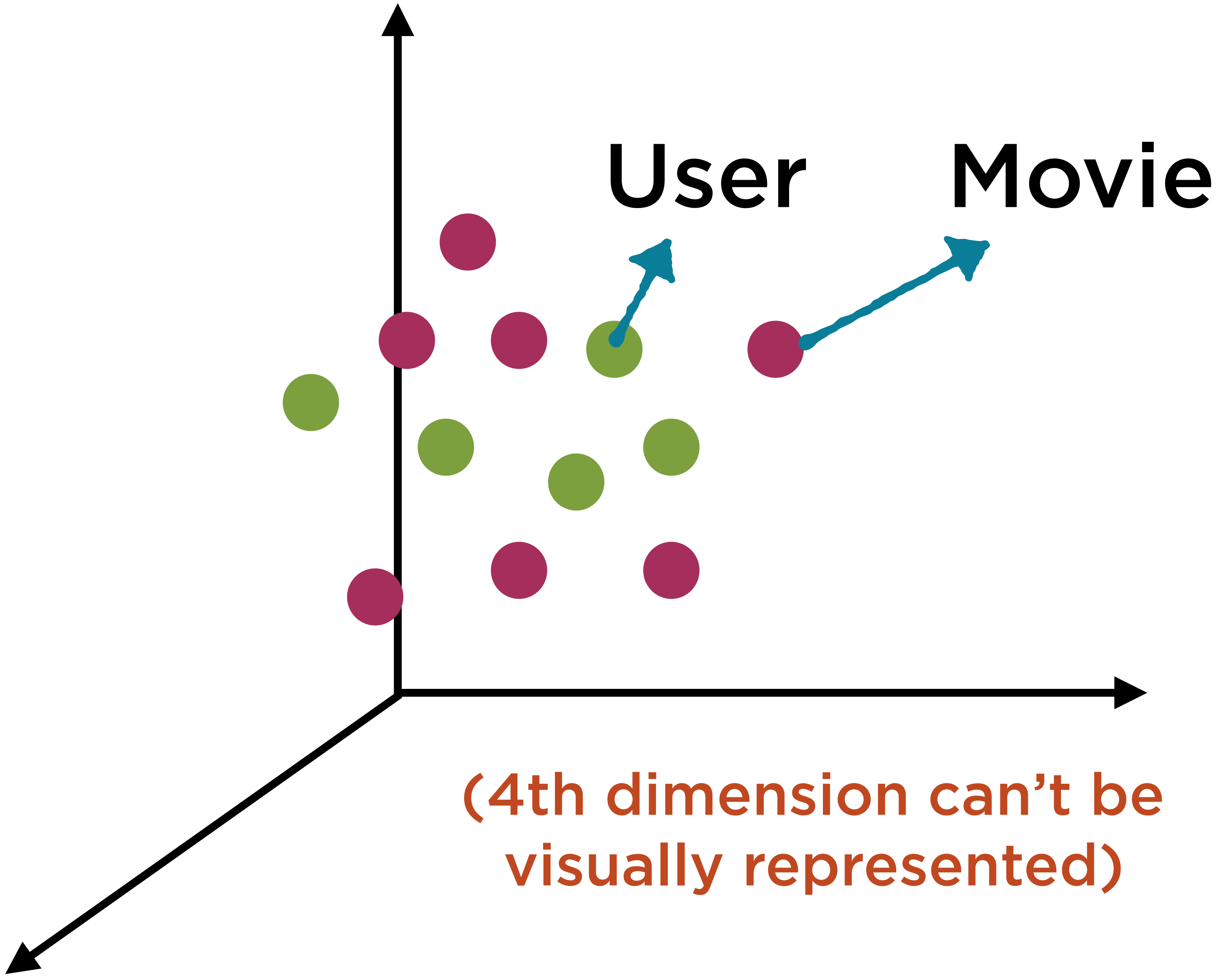
Cast Popularity

Recency of release

Every user and every movie is represented using 4 numbers

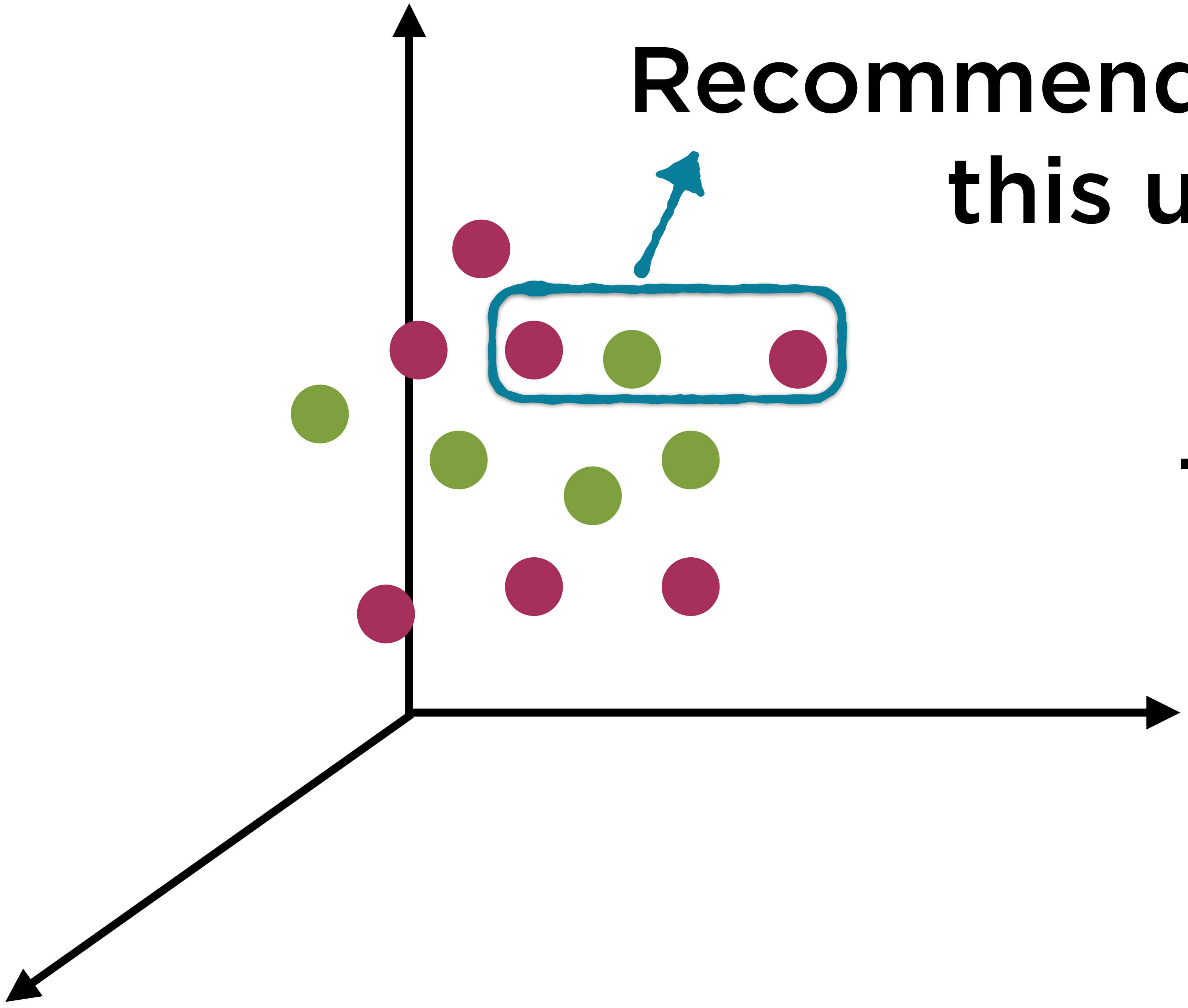
Points in 4-D Space

Points in 4-D Space



(4th dimension can't be visually represented)

Points in 4-D Space



Recommendations for this user

The "nearest" movies

Hidden Factors for Movies

- Genre
- Commercial Appeal
- Cast Popularity
- Recency of release

The key step in this approach

Representing users and movies in terms of influencing factors

Assumptions

1. We know which factors influence user's preferences

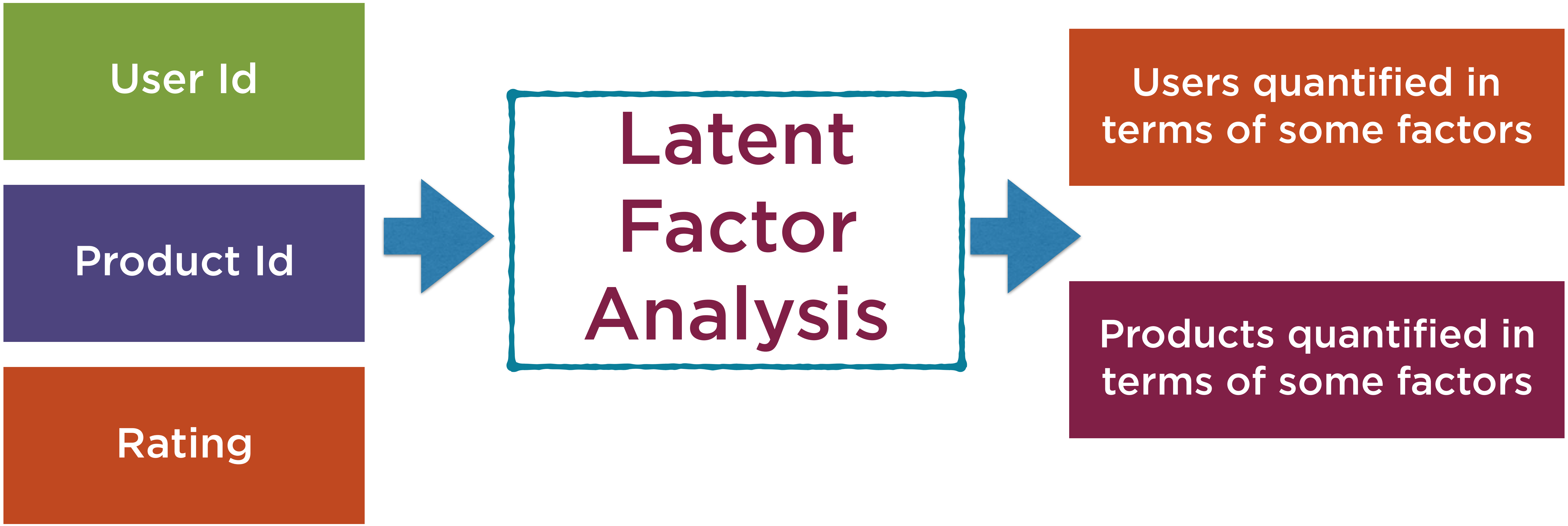
2. We can quantify those factors for each user

3. We can quantify those factors for each movie

Latent Factor Analysis Techniques

Attempt to do all of this, using just

User Id	Product Id	Rating


User Id
Product Id
Rating
Latent Factor Analysis
Users quantified in terms of some factors
Products quantified in terms of some factors

The factors are **not known beforehand**

Users quantified in terms of some factors

Products quantified in terms of some factors

Once they are identified, they might turn out to be

Factors with meaning like genre, cast popularity

Abstract factors with no real life meaning

Users quantified in terms of some factors

Products quantified in terms of some factors

To identify factors, represent the training data as a matrix

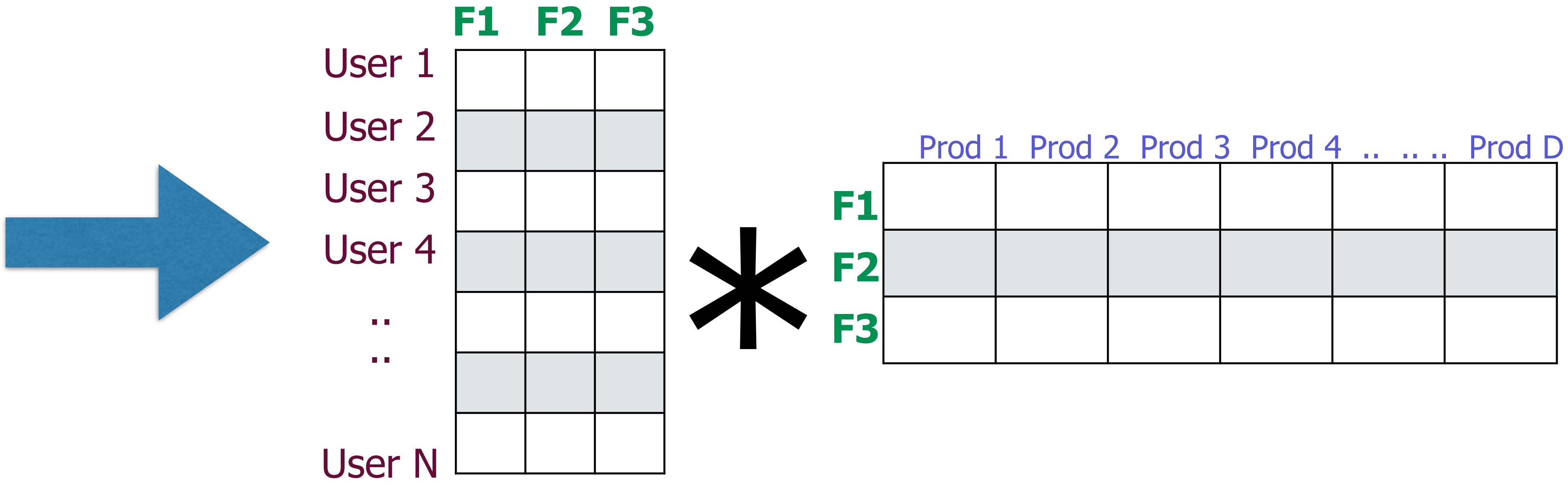
User Id

Product Id

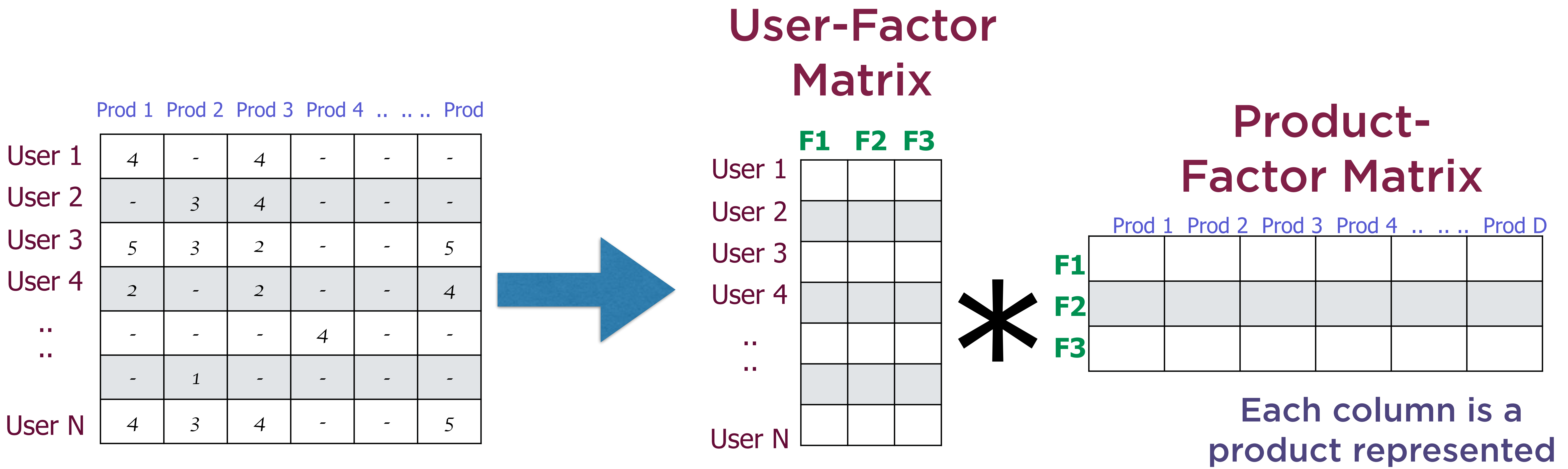
Rating

	Prod 1	Prod 2	Prod 3	Prod 4	.. .. ..	Prod D
User 1	4	-	4	-	-	-
User 2	-	3	4	-	-	-
User 3	5	3	2	-	-	5
User 4	2	-	2	-	-	4
:	-	-	-	4	-	-
:	-	1	-	-	-	-
User N	4	3	4	-	-	5

	Prod 1	Prod 2	Prod 3	Prod 4	.. .. ..	Prod D
User 1	4	-	4	-	-	-
User 2	-	3	4	-	-	-
User 3	5	3	2	-	-	5
User 4	2	-	2	-	-	4
..	-	-	-	4	-	-
..	-	1	-	-	-	-
User N	4	3	4	-	-	5

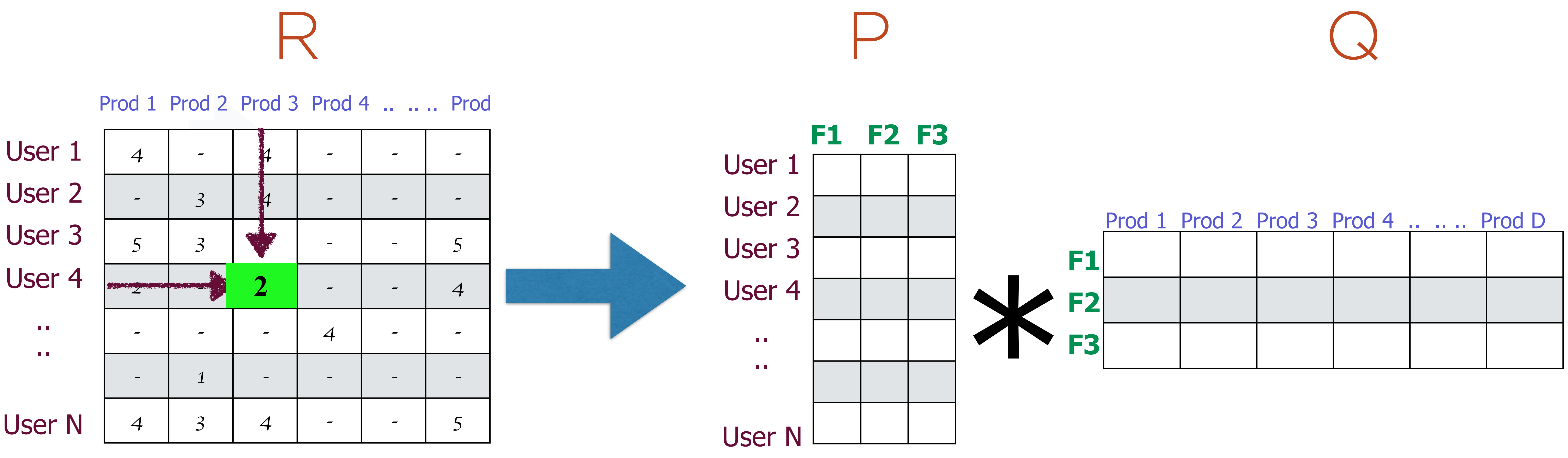


Latent Factor analysis breaks this down

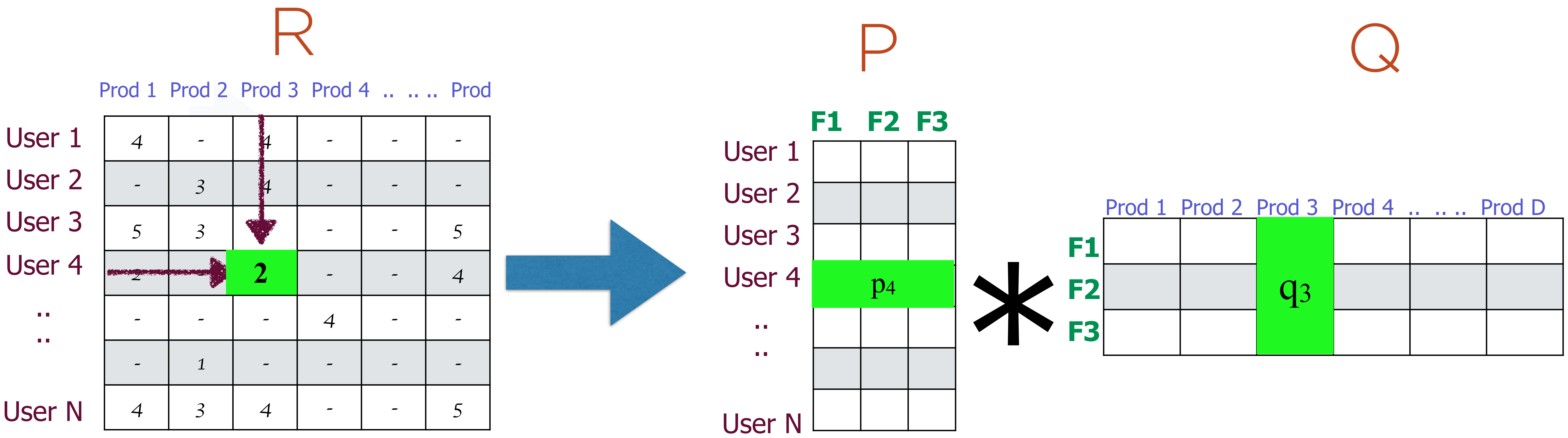


Each row is a user represented in terms of the hidden factors

Each column is a product represented in terms of the hidden factors



r_{43} A Rating by User 4 for Product 3


R
Prod 1 Prod 2 Prod 3 Prod 4 .. .. .. Prod
User 1
User 2
User 3
User 4
User N
4 - 4 - - -
- 3 4 - - -
5 3 - - 5
2 2 - - 4
- - - 4 - -
- 1 - - - -
4 3 4 - - 5
P
F1 F2 F3
User 1
User 2
User 3
User 4
User N
p4
Q
Prod 1 Prod 2 Prod 3 Prod 4 .. .. .. Prod D
F1
F2
F3
q3

$$r_{43} = p_4 . q_3$$

$$r_{ui} = p_u \cdot q_i$$

The set of equations that represent the matrix decomposition

$$r_{ui} = p_u \cdot q_i$$

Solve this set of equations to have all users, products quantified in terms of factors

Latent Factor Analysis

The problem is expressed as

Find that set of factor vectors

p_u (For each user u)

q_i (For each item i)

Such that the total error is minimized

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2$$

$$\min_{q^*,p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2$$

Alternating Least Squares

is a technique to minimize this error

Alternating Least Squares

Concentrate on one rating r_{ui}

The equation to solve

$$(r_{ui} - q_i^T p_u)^2 = 0$$

There are 2 variables i.e. p_u and q_i

Alternating Least Squares

$$(r_{ui} - q_i^T p_u)^2 = 0$$

What if we fixed the value of p_u ?

We are left with a quadratic equation for the value of q_i

What if we fixed the value of q_i to the solved value?

We are left with a quadratic equation for the value of p_u

Repeat until the values p_u and q_i converge

Alternating Least Squares

The choice of the number of hidden factors is left to the user

Regularization

In general in Machine Learning, we want to find simpler models

If a simpler model can explain the same relationship, then we would like to choose that

Alternating Least Squares

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

Adding a regularization term

Penalizes models with higher number of factors

Alternating Least Squares

When you implement ALS

Number of factors, Lambda

are parameters to experiment with

Demo

Implement Alternating Least Squares in Python

Find Movie Recommendations for users using the MovieLens data set

Summary

Understand the role of Personalized Recommendations

Predict ratings using Collaborative Filtering

Find hidden factors that influence ratings

Implement Movie Recommendations in Python