Recommendation System

推荐系统

You need to recommend a financial product to your users. There are countless financial products available for users to choose from. Which financial product would you recommend to your users? Why?

你要推荐一款理财产品给你的用户。有无数种理财产品供用户选择，你将推荐哪一款理财产品给你的用户呢？为什么？

Here are some movie ratings.

这里是一些电影评分

	Movie 1	**Movie 2**	**Movie 3**	**Movie 4**
Alice	4	4		1
Bob		2	2	3
Carol	1	5	3	
Dennis	3		4	1
Emma	5	2	1	4
Flora	3	1		5

Predict Alice's rating for movie 3. What's your reason?

预测 Alice 对第三部电影的评分。你的理由是什么?

	Movie 1	**Movie 2**	**Movie 3**	**Movie 4**
Alice	4	4	???	1
Bob		2	2	3
Carol	1	5	3	
Dennis	3		4	1
Emma	5	2	1	4
Flora	3	1		5

Recommendation is everywhere!

推荐无处不在！

amazon.com

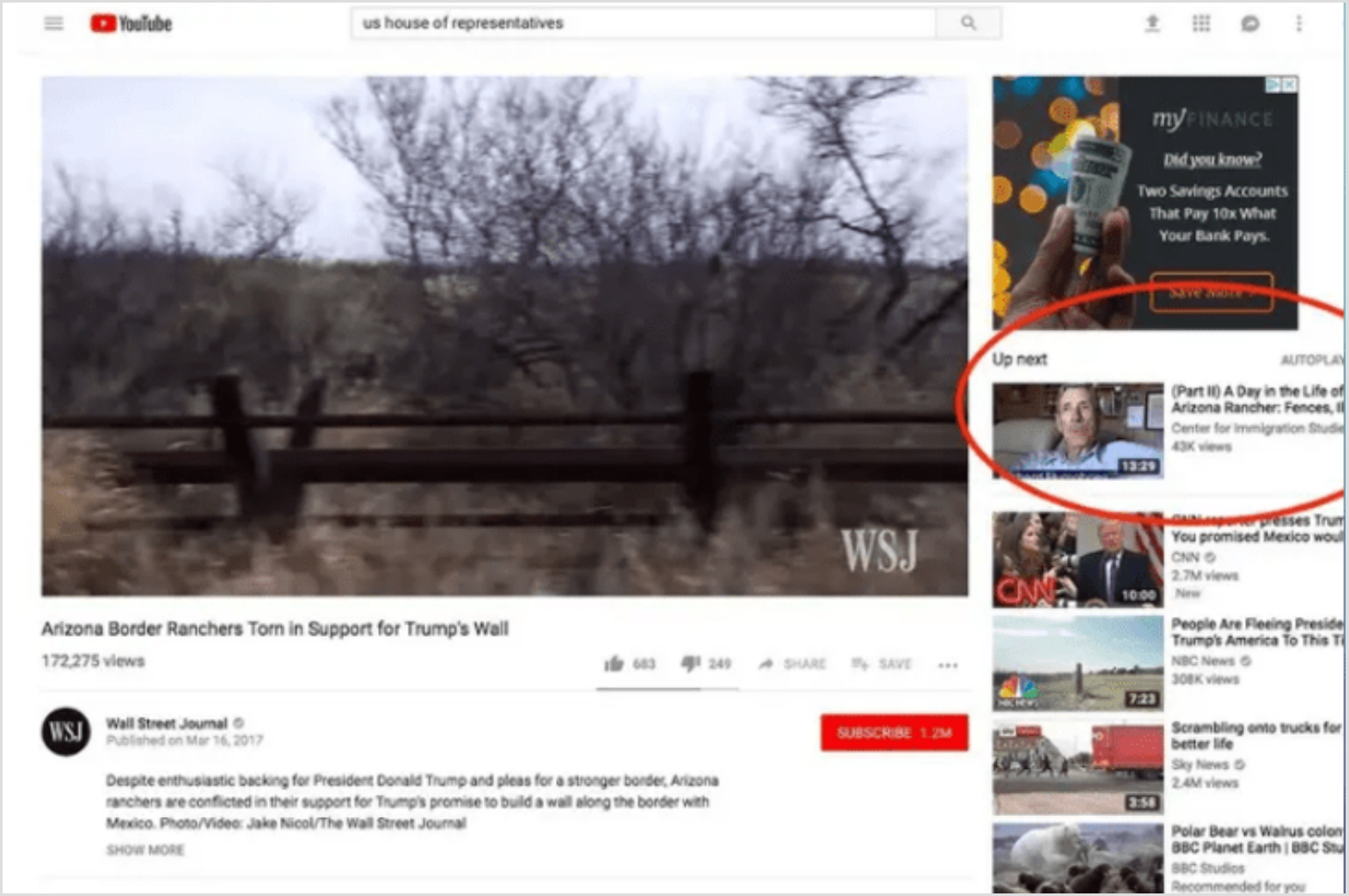
Recommended for You

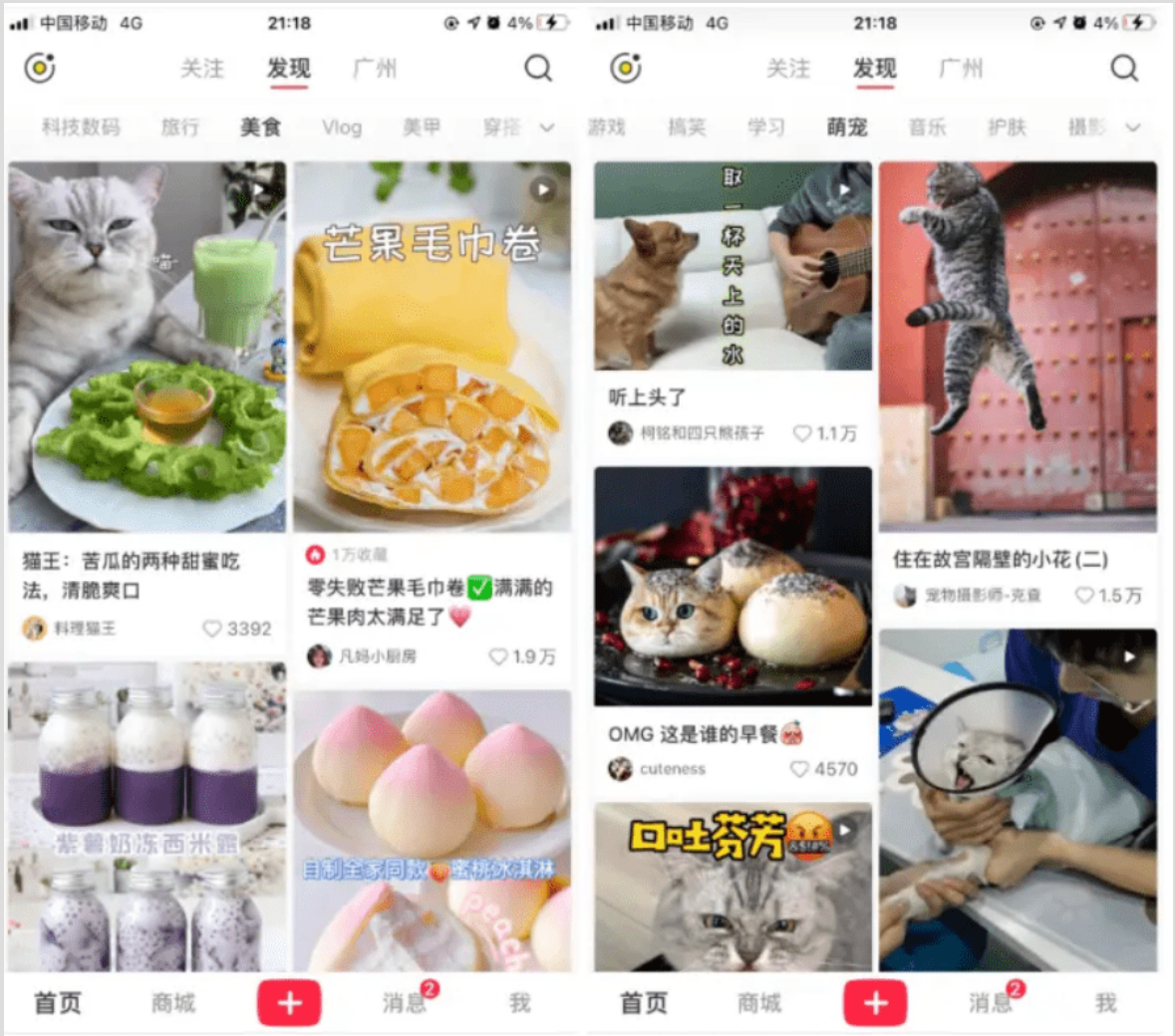
Amazon.com has new recommendations for you based on items you purchased or told us you own.

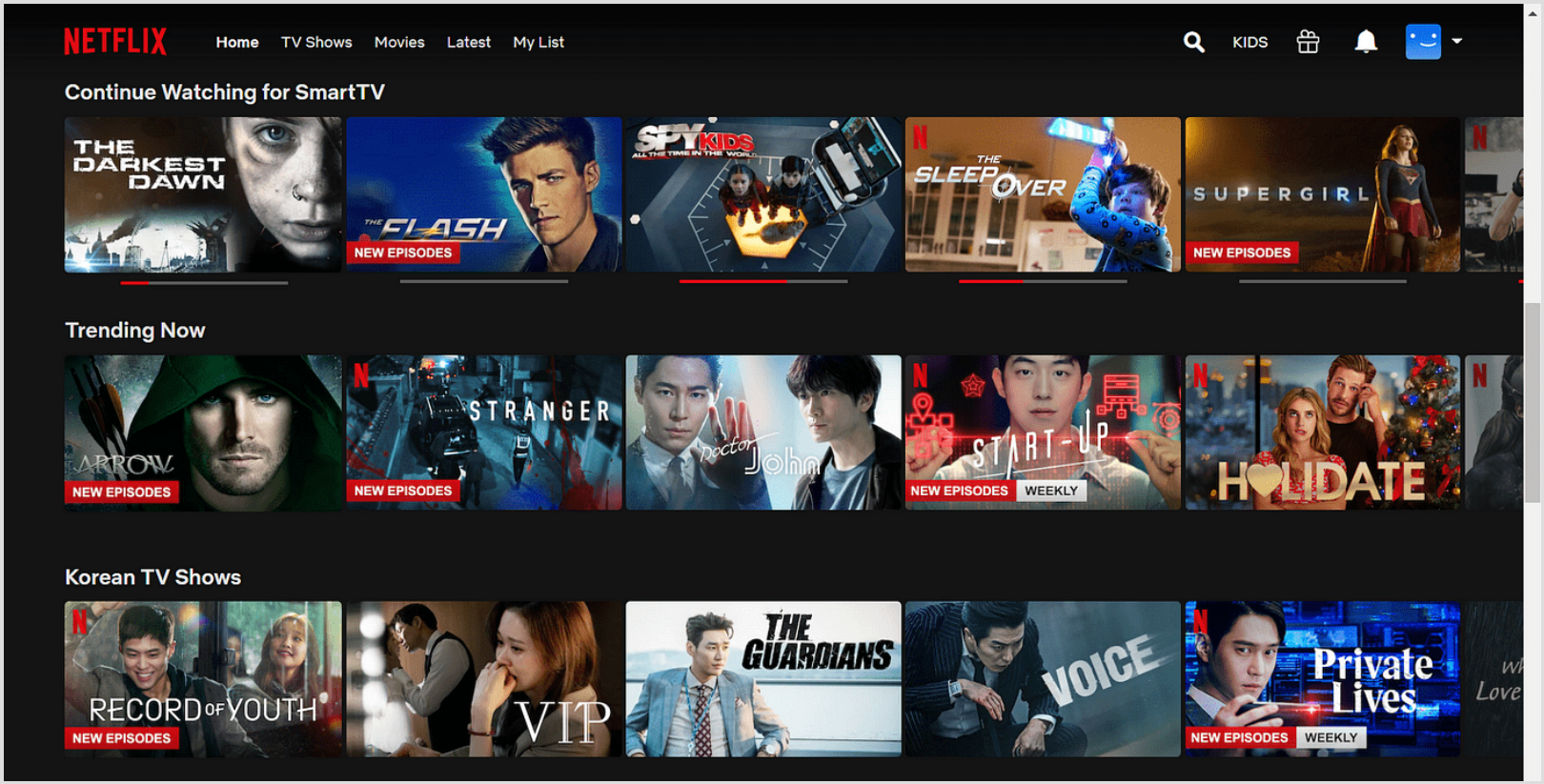
Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop

Google Apps Administrator Guide: A Private-Label Web Workspace

Googlepedia: The Ultimate Google Resource (3rd Edition)


YouTube
us house of representatives
WSJ
Arizona Border Ranchers Torn in Support for Trump's Wall
172,275 views
683
249
SHARE
SAVE
Wall Street Journal
Published on Mar 16, 2017
SUBSCRIBE 1.2M
Despite enthusiastic backing for President Donald Trump and pleas for a stronger border, Arizona ranchers are conflicted in their support for Trump's promise to build a wall along the border with Mexico. Photo/Video: Jake Nicol/The Wall Street Journal
SHOW MORE
myFINANCE
Did you know?
Two Savings Accounts That Pay 10x What Your Bank Pays.
Up next
AUTOPLAY
(Part II) A Day in the Life of Arizona Rancher: Fences, II
Center for Immigration Studies
43K views
13:29
You promised Mexico woul
CNN
2.7M views
New
10:00
People Are Fleeing Preside Trump's America To This Ti
NBC News
308K views
7:23
Scrambling onto trucks for better life
Sky News
2.4M views
3:58
Polar Bear vs Walrus colon BBC Planet Earth | BBC Stu
BBC Studios
Recommended for you


中国移动 4G
21:18
4%
关注
发现
广州
科技数码
旅行
美食
Vlog
美甲
穿搭
芒果毛巾卷
猫王：苦瓜的两种甜蜜吃法，清脆爽口
料理猫王
3392
1万收藏
零失败芒果毛巾卷✅满满的芒果肉太满足了❤️
凡妈小厨房
1.9万
紫薯奶冻西米露
自制全家同款🍑蜜桃冰淇淋
peach
首页
商城
消息
我
关注
发现
广州
游戏
搞笑
学习
萌宠
音乐
护肤
摄影
取一杯天上的水
听上头了
柯铭和四只熊孩子
1.1万
住在故宫隔壁的小花(二)
宠物摄影师-克意
1.5万
OMG 这是谁的早餐
cuteness
4570
口吐芬芳
首页
商城
消息
我


NETFLIX
Home
TV Shows
Movies
Latest
My List
KIDS
Continue Watching for SmartTV
THE DARKEST DAWN
THE FLASH
NEW EPISODES
SPY KIDS
THE SLEEPOVER
SUPERGIRL
NEW EPISODES
Trending Now
ARROW
NEW EPISODES
STRANGER
NEW EPISODES
doctor John
START-UP
NEW EPISODES
WEEKLY
HOLIDATE
Korean TV Shows
RECORD OF YOUTH
NEW EPISODES
VIP
THE GUARDIANS
VOICE
Private Lives
NEW EPISODES
WEEKLY

BBC
Home News Sport Business Innovation Culture Travel Earth Video Live
Netflix: How did it know I was bi before I did?
13 August 2023
By Ellie House, BBC Long Form Audio
Share

The Importance of Recommendation

- Netflix: 2/3 of the movies watched are recommended.
- Google News: recommendations generate 38% more click-throughs.
- Amazon: 35% sales from recommendations.
- ChoiceStream: 28% of the people would buy more music if they found what they liked.

推荐的重要性

- Netflix: 观看的电影中有2/3是根据推荐的。
- Google News: 推荐产生了比例高达38%的点击率。
- 亚马逊: 35%的销售额来自推荐。
- ChoiceStream: 如果找到自己喜欢的音乐，28%的人会购买更多。

How to recommend?

A recommendation system must have three inputs:

- Items to be recommended: songs, movies, products, restaurants etc. (often many thousands)
- Users of the items: watchers, listeners, purchasers, shoppers etc. (often many millions)
- Feedback of users on items: 5-star ratings, upvotes/downvotes, clicking “next” or “skipping the ad”, purchases or clicks.

如何进行推荐?

一个推荐系统必须有三个输入:

- 要推荐的物品: 歌曲、电影、产品、餐馆等（通常有成千上万个）
- 使用这些物品的用户: 观看者、听众、购买者、购物者等（通常有数百万人）
- 用户对物品的反馈: 5星评级、赞成/反对、点击“下一个”或“跳过广告”、购买或点击等。

Collaboratives Filtering

Collaborative filtering is not something new. We have done it in many places in the past. Here are a few examples:

- Bestseller list for books
- Top 50 music list
- The “recent returns” shelf at libraries

The intuition behind: People’s tastes are correlated.

协同过滤

协同过滤并非什么新鲜事物。我们在过去的许多地方都使用过它。以下是一些示例：

- 畅销书榜单
- 前50名音乐榜单
- 图书馆的“最近归还”书架

背后的直觉是：人们的口味是相关的。

Collaboratives Filtering

However, in the above examples, recommendations are not personalized, i.e., everybody receives the same recommendation. How to make recommendations personalized?

The intuition: If Alice and Bob both like X and Alice also likes Y, then Bob is more likely to like Y, especially when Alice and Bob know each other.

协同过滤

然而，在上述例子中，推荐并不是个性化的，即每个人都会收到相同的推荐。如何使推荐个性化?

直觉：如果 Alice 和 Bob 都喜欢 X，并且 Alice 还喜欢Y，那么 Bob 更有可能喜欢 Y，尤其是当 Alice 和 Bob 互相认识时。

Suppose that you want to recommend a movie to Emma, which movie will you recommend?

	1	**2**	**3**	**4**	**5**	**6**
Alice	2			4	5	
Bob	5		4			1
Carol			5		2	
Dennis		1		5		4
Emma			4			2
Flora	4	5		1		

假设你想向 Emma 推荐一部电影，你会推荐哪部电影?

	1	**2**	**3**	**4**	**5**	**6**
Alice	2			4	5	
Bob	5		4			1
Carol			5		2	
Dennis		1		5		4
Emma			4			2
Flora	4	5		1		

User-Based Collaborative Filtering: The Neighbourhood Method

基于用户的协同过滤： 邻域方法

Step 1: Find all the movies rated by Emma before, we get movies 3 and 6

步骤 1：找到 Emma 之前评过的所有电影，我们得到了电影 3 和 6。

	1	**2**	**3**	**4**	**5**	**6**
Alice	2			4	5	
Bob	5		4			1
Carol			5		2	
Dennis		1		5		4
Emma			4			2
Flora	4	5		1		

Step 2: Identify other users that have rated the same movie, we get Bob, Carol, and Dennis

步骤 2：识别其他评过同一部电影的用户，我们得到了Bob, Carol, 和 Dennis

	1	**2**	**3**	**4**	**5**	**6**
Alice	2			4	5	
Bob	5		4			1
Carol			5		2	
Dennis		1		5		4
Emma			4			2
Flora	4	5		1		

Step 3: Compare the similarity between Emma and her "neighbors" to see who are close to Emma.

步骤 3：比较 Emma 与她的"邻居"之间的相似性，以确定谁与 Emma 接近。

	1	**2**	**3**	**4**	**5**	**6**
Alice	2			4	5	
Bob	5		4			1
Carol			5		2	
Dennis		1		5		4
Emma			4			2
Flora	4	5		1		

Step 4: Select the top k most similar neighbors and use their average ratings to predict Emma's rating.

步骤 4：选择最相似的前 k 个邻居，并使用他们的平均评分来预测艾玛的评分。

	1	**2**	**3**	**4**	**5**	**6**
Alice	2			4	5	
Bob	5		4			1
Carol			5		2	
Dennis		1		5		4
Emma			4			2
Flora	4	5		1		

Item-Based Collaborative Filtering

基于物品的协同过滤

Item-based collaborative filtering

基于物品的协同过滤

Suppose that we are predicting the who will like movie 5.

假设我们正在预测谁会喜欢电影 5

Step 1: Who have rated movie 5 before? We get Alice and Carol.

步骤 1：谁之前评过电影 5？我们得到了 Alice 和 Carol

	1	**2**	**3**	**4**	**5**	**6**
Alice	2			4	5	
Bob	5		4			1
Carol			5		2	
Dennis		1		5		4
Emma			4			2
Flora	4	5		1		

Step 2: Identify other movies that have rated the same users, we get movies 1 and 3.

步骤 2：识别这些用户评过的其他电影，我们得到了电影 1 和 3

	1	2	3	4	5	6
Alice	2			4	5	
Bob	5		4			1
Carol			5		2	
Dennis		1		5		4
Emma			4			2
Flora	4	5		1		

Step 3: Compare the similarity between movie 5 and its "neighbors" to see which movie is close to movie 5.

步骤 3：比较电影 5 与其"邻居"之间的相似性，以确定哪部电影接近电影 5

	1	2	3	4	5	6
Alice	2			4	5	
Bob	5		4			1
Carol			5		2	
Dennis		1		5		4
Emma			4			2
Flora	4	5		1		

Step 4: Select the top k most similar neighbors and use their average ratings to predict movie 5's rating.

步骤 4：选择最相似的前 k 个邻居，并使用他们的平均评分来预测电影 5 的评分。

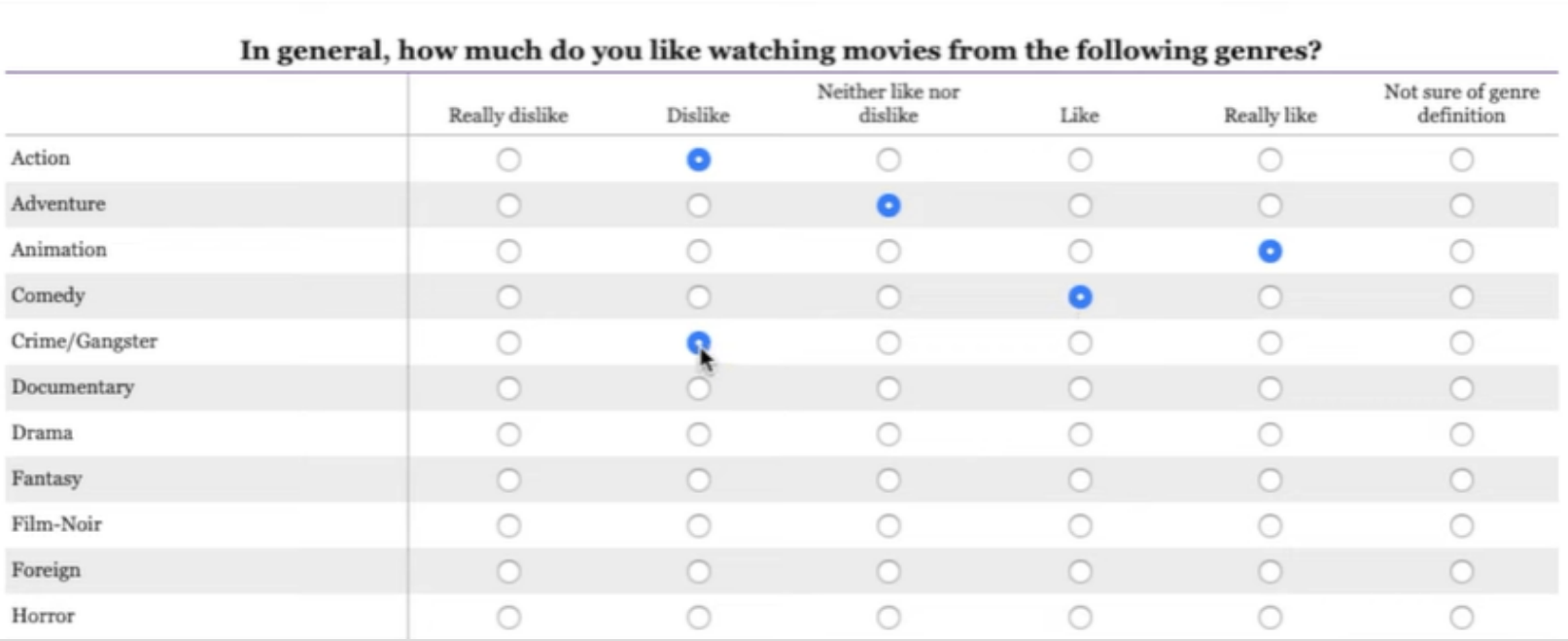
	1	2	3	4	5	6
Alice	2			4	5	
Bob	5		4			1
Carol			5		2	
Dennis		1		5		4
Emma			4			2
Flora	4	5		1		

Model-based Collaborative Filtering

基于模型的协同过滤

What did Netflix do to make recommendations?

Netflix 早年是如何进行推荐的?



In general, how much do you like watching movies from the following genres?

	Really dislike	Dislike	Neither like nor dislike	Like	Really like	Not sure of genre definition
Action	○	●	○	○	○	○
Adventure	○	○	●	○	○	○
Animation	○	○	○	○	●	○
Comedy	○	○	○	●	○	○
Crime/Gangster	○	●	○	○	○	○
Documentary	○	○	○	○	○	○
Drama	○	○	○	○	○	○
Fantasy	○	○	○	○	○	○
Film-Noir	○	○	○	○	○	○
Foreign	○	○	○	○	○	○
Horror	○	○	○	○	○	○

Issues with this approach:

- It is takes users much effort to complete the questionnaire, and users do not always bother to answer the questions carefully and seriously.
- Sometimes users themselves cannot clearly state their preferences.
- Maybe there are some hidden factors that are not captured by the questionnaire, e.g., the duration of the movie.

该方法的问题：

- 用户完成问卷需要付出很大努力，用户并不总是愿意仔细认真地回答问题。
- 有时用户自己也无法清楚地表达自己的偏好。
- 可能存在一些问卷未能捕捉到的隐含因素，例如电影的时长。

The Matrix Factorization Method

矩阵分解方法

Matrix Factorization

Here, we assume that each movie has a number of “latent” or “hidden” factors that affect user preferences. Examples of the factors include the length of the movie, the amount of actions in the movie, the seriousness of the movie, the orientation of the movie for children etc.

The factors are “latent” or “hidden,” implying that we do not know which these factors are, and we do not need to know either.

矩阵分解

在这里，我们假设每部电影都有一些影响用户偏好的“潜在”或“隐含”因素。这些因素的例子包括电影的时长、电影中的动作量、电影的严肃程度、电影是否适合儿童等。

这些因素是“潜在”或“隐含”的，意味着我们不知道这些因素具体是什么，也不需要知道。

Matrix Factorization

Each user also has his or her own preference for each factor. For instance, some users prefer long movies over short movies, and some users prefer to have more actions in their movie. If we know the preferences of a user and a movie's attribute values, we can match the movie with the user to see whether the user will like the movie.

矩阵分解

每个用户对每个因素也有自己的偏好。例如，有些用户更喜欢长电影而不是短电影，有些用户则更喜欢电影中有更多动作。如果我们知道用户的偏好以及电影的属性值，就可以将电影与用户匹配，以判断用户是否会喜欢这部电影。

Matrix Factorization

For instance, suppose that a movie is long and contains a lot of actions. We also know that

- Alice likes short movies and hates action movies,
- Bob prefers long movies and enjoys action movies.

Then, we can predict that Alice will hate the movie and Bob will like the movie.

矩阵分解

例如，假设一部电影很长且包含很多动作。我们还知道：

- Alice 喜欢短电影并讨厌动作电影，
- Bob 更喜欢长电影并喜欢动作电影。

那么，我们可以预测 Alice 会讨厌这部电影，而 Bob 会喜欢这部电影。

Mathematically, our model is as follows:

$$\begin{aligned}\text{Your rating} = &\text{ Your preference for length} \times \text{Movie's length} \\ &+ \text{Your preference for action} \times \text{Movie's amount of action}\end{aligned}$$

Suppose that the movie's length is 1 and amount of action is 2. Alice's preference for length is 0.5, for action is 0; Bob's preference for length is 1, for action is 1.5. We can predict:

- Alice's rating: $0.5 \times 1 + 0 \times 2 = 0.5$.
- Bob's rating: $1 \times 1 + 1.5 \times 2 = 4$.

在数学上，我们的模型如下：

$$\text{你的评分} = \text{你对市场的偏好} \times \text{电影时长} + \text{你对动作的偏好} \times \text{电影动作含量}$$

假设这部电影的时长为 1，动作量为 2。Alice 对时长的偏好为 0.5，对动作的偏好为 0；Bob 对时长的偏好为 1，对动作的偏好为 1.5。我们可以预测：

- Alice 的评分: $0.5 \times 1 + 0 \times 2 = 0.5$.
- Bob 的评分: $1 \times 1 + 1.5 \times 2 = 4$.

Let's generalize the above discussion. Suppose that Alice, Bob, Carol's preferences for length and action are as follows:

	Length	**Action**
Alice	0.5	0
Bob	1	1.5
Carol	1.5	0.5

There are two movies, whose length and action values are

	Length	**Action**
1	1	2
2	0	3

让我们对上述讨论进行概括。假设 Alice, Bob, Carol 对时长和动作的偏好如下：

	Length	**Action**
Alice	0.5	0
Bob	1	1.5
Carol	1.5	0.5

有两部电影，它们的时长和动作值分别为：

	Length	**Action**
1	1	2
2	0	3

We can multiply the two matrix to get user-movie ratings:

我们可以将这两个矩阵相乘以获得用户-电影评分：

$$\begin{bmatrix} 0.5 & 0 \\ 1 & 1.5 \\ 1.5 & 0.5 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 4 & 4.5 \\ 2.5 & 1.5 \end{bmatrix}$$

In other words, our prediction is as follows.

	Movie 1	**Movie 2**
Alice	0.5	0.0
Bob	4.0	4.5
Carol	2.5	1.5

Overall, if we know the user matrix and the movie matrix, we can multiply the two to get the user-movie rating matrix. The issue is: We do not yet know the user matrix and the movie matrix.

换句话说，我们的预测如下

	Movie 1	**Movie 2**
Alice	0.5	0.0
Bob	4.0	4.5
Carol	2.5	1.5

总体来说，如果我们知道用户矩阵和电影矩阵，我们可以将两者相乘以获得用户-电影评分矩阵。问题是：我们尚不知道用户矩阵和电影矩阵。

But how to get the user matrix and the movie matrix?

The short answer is, we can guess. We guess different user matrices and movie matrices, and see if the predicted rating matrix is close to the actual rating given by users. When the two are close enough, we can use the two matrices to construct the new rating matrix. This is known as matrix factorization.

但是如何获得用户矩阵和电影矩阵呢?

简短的回答是，我们可以进行猜测。我们猜测不同的用户矩阵和电影矩阵，然后查看预测的评分矩阵是否接近用户给出的实际评分。当两者足够接近时，我们可以使用这两个矩阵来构建新的评分矩阵。这被称为矩阵分解。

We can do better than guessing. There are some advanced statistical methods for estimating the user matrix and movie matrix, but this is beyond the scope of our class. If you are interested, you can search for "stochastic gradient descent."

我们可以做得比单纯猜测更好。有一些高级统计方法可以用来估计用户矩阵和电影矩阵，但这超出了我们课程的范围。如果你感兴趣，可以搜索"随机梯度下降"。

The matrix factorization algorithm

矩阵分解算法

```
1  matrix_factorization <- function(R, P, Q, K, steps=5000, alpha=0.0002,
   beta=0.02) {
2    Q <- t(Q)
3    for (step in 1:steps) {
4      for (i in 1:nrow(R)) {
5        for (j in 1:ncol(R)) {
6          if (R[i, j] > 0) {
7            eij <- R[i, j] - sum(P[i,] * Q[,j])
8            for (k in 1:K) {
9              P[i, k] <- P[i, k]+alpha*(2*eij*Q[k, j] - beta*P[i, k])
10             Q[k, j] <- Q[k, j]+alpha*(2*eij*P[i, k] - beta*Q[k, j])
11           }}}}
12     eR <- P %*% Q
13     e <- 0
14     for (i in 1:nrow(R)) {
15       for (j in 1:ncol(R)) {
16         if (R[i, j] > 0) {
17           e <- e + (R[i, j] - sum(P[i,] * Q[,j]))^2
18             for (k in 1:K) {
19             e <- e + (beta/2) * (P[i, k]^2 + Q[k, j]^2)
20           }}}}
21     if (e < 0.001){break}}
22   return(list(P = P, Q = t(Q)))
23 }
```

```r
set.seed(123)
R <- matrix(c(5, 3, 0, 1,
              4, 0, 0, 1,
              1, 1, 0, 5,
              1, 0, 0, 4,
              0, 1, 5, 4,
              2, 1, 3, 0), nrow = 6, ncol = 4, byrow = TRUE)
# N: num of User
N <- nrow(R)
# M: num of Movie
M <- ncol(R)
# Num of Features
K <- 2
P <- matrix(runif(N * K), nrow = N, ncol = K)
Q <- matrix(runif(M * K), nrow = M, ncol = K)
result <- matrix_factorization(R, P, Q, K)
nP <- result$P
nQ <- result$Q
nR <- nP %*% t(nQ)
print(nP)
print(nQ)
print(nR)
```

Back to our example...

回到我们的例子……

	Movie 1	**Movie 2**	**Movie 3**	**Movie 4**
Alice	4	4		1
Bob		2	2	3
Carol	1	5	3	
Dennis	3		4	1
Emma	5	2	1	4
Flora	3	1		5

$$\begin{bmatrix} 4 & 4 & ? & 1 \\ ? & 2 & 2 & 3 \\ 1 & 5 & 3 & ? \\ 3 & ? & 4 & 1 \\ 5 & 2 & 1 & 4 \\ 3 & 1 & ? & 5 \end{bmatrix} \approx \begin{bmatrix} 1.71 & 0.74 \\ 0.84 & 1.35 \\ 1.92 & -0.69 \\ 2.05 & 0.46 \\ 0.70 & 1.95 \\ 0.13 & 1.93 \end{bmatrix} \times \begin{bmatrix} 1.24 & 2.44 & 1.77 & -0.20 \\ 1.83 & 0.14 & 0.10 & 2.32 \end{bmatrix}$$

$$= \begin{bmatrix} 3.47 & 4.28 & 3.09 & 1.37 \\ 3.52 & 2.26 & 1.63 & 2.97 \\ 1.14 & 4.63 & 3.34 & -1.99 \\ 3.41 & 5.09 & 3.67 & 0.66 \\ 4.48 & 1.99 & 1.43 & 4.40 \\ 3.69 & 0.60 & 0.43 & 4.46 \end{bmatrix}$$

https://www.youtube.com/embed/n3RKsY2H-NE?enablejsapi=1

How to recommend financial products using the methods above?

如何用上述方法推荐理财产品?

The Cold Start Problem

冷启动问题

The cold start problem

The collaborative filtering algorithm works very well in generally, yet it suffers from the issue of cold start. Recall that in the recommendation algorithm, we need to know the users' past interaction with the items to make recommendations. However, if a user or an item is completely new without any historical interactions, what would you do to make recommendations?

冷启动问题

协同过滤算法通常效果很好，但它面临着冷启动问题。回想一下，在推荐算法中，我们需要了解用户与项目的过去互动才能进行推荐。然而，如果一个用户或一个项目是全新的，没有任何历史互动，你会如何进行推荐呢？

Social Network Analysis

社交网络分析

Lenddo, a Singaporean start-up, helps financial institutions collect users' social network data. But why?

Lenddo，一家新加坡初创公司，帮助金融机构收集用户的社交网络数据。但为什么呢?

Obesity is an epidemic.

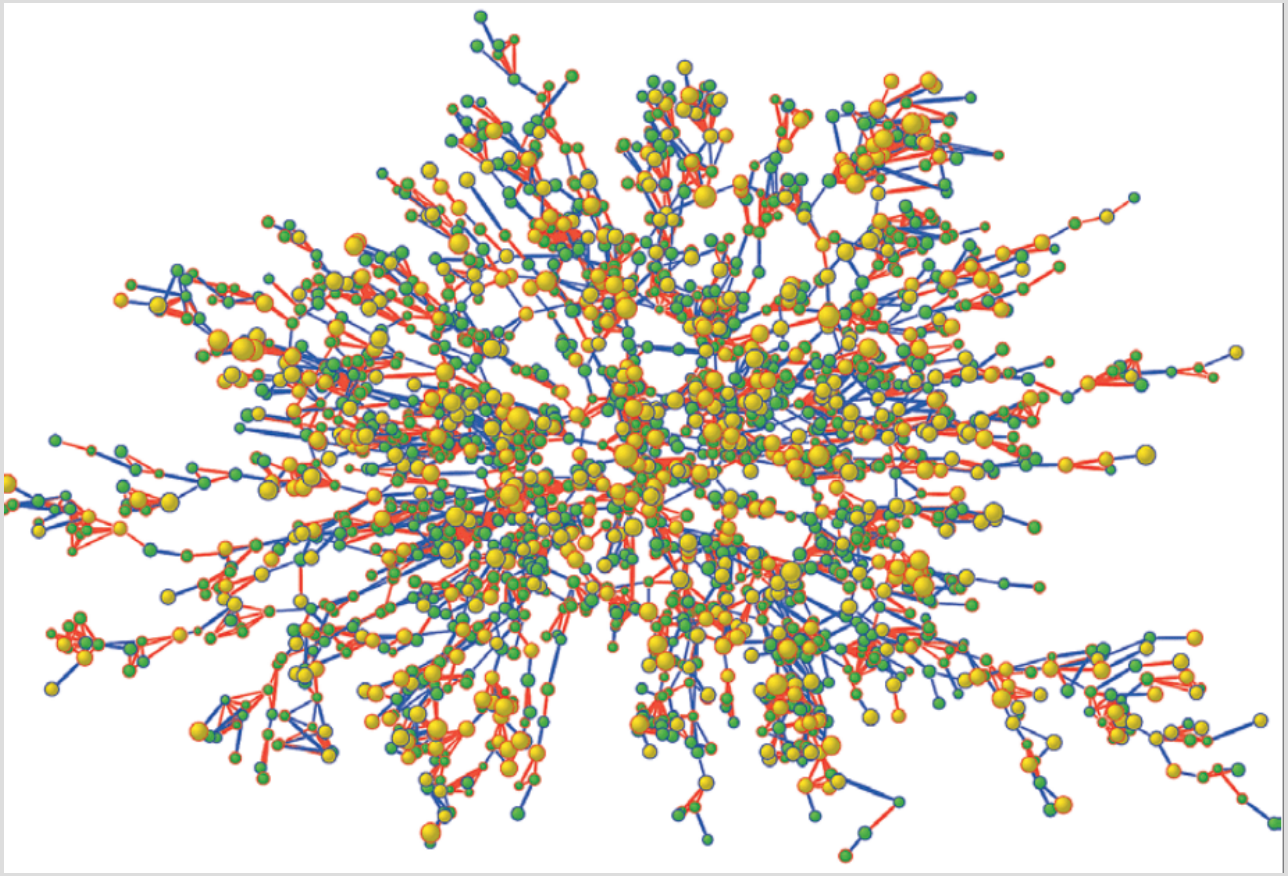
肥胖是一种流行病

The NEW ENGLAND JOURNAL of MEDICINE

SPECIAL ARTICLE

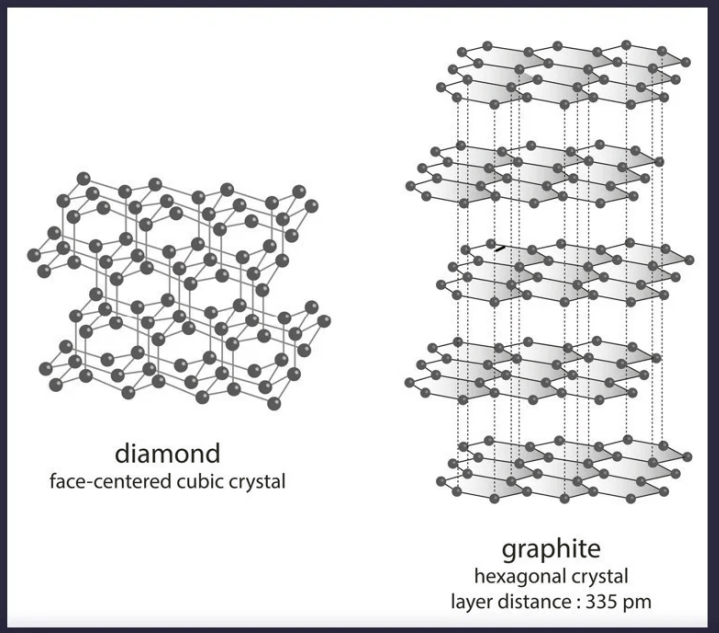
The Spread of Obesity in a Large Social Network over 32 Years

Nicholas A. Christakis, M.D., Ph.D., M.P.H., and James H. Fowler, Ph.D.



Node: individual; edge: connections; size of node: body mass index; yellow: obesity (i.e., $BMI > 30$.)

节点: 个体; 边: 连接; 节点大小: 身体质量指数; 黄色: 肥胖(即 $BMI>30$)



Network structure makes the difference.

网络结构决定差异

Key Metrics of a Social Network

Individual: Has meaning independently of social network
You live in Shenzhen

Connection: You are close friends with 10 people at CCB

Whole Network: On average, employees know each other within 4 steps

Connection can be directed (e.g., follower and followee) or undirected (e.g., classmates)

社交网络的关键指标

个体：在社交网络之外独立存在意义(你住在深圳)

连接：你与建设银行的10个好友关系密切

整体网络：平均而言，员工之间相互认识的距离为4步

连接可以是有向的[例如，关注者和被关注者]或无向的[例如，同班同学]

Nodes and Edges

Vertex/Node: an end point, often a person

Edge/Link: What connects up the nodes, e.g., a relationship

Maximum number of edges in group of size $N(N-1)/2$.

- Where everyone connects to everyone else
- If undirected (my friends also have me as a friend)

节点和连接

节点: 一个端点，通常是一个人。

连接或边: 连接节点的是什么，例如，一种拥有 N 个节点的系统中最大边数为 $N(N-1)/2$.

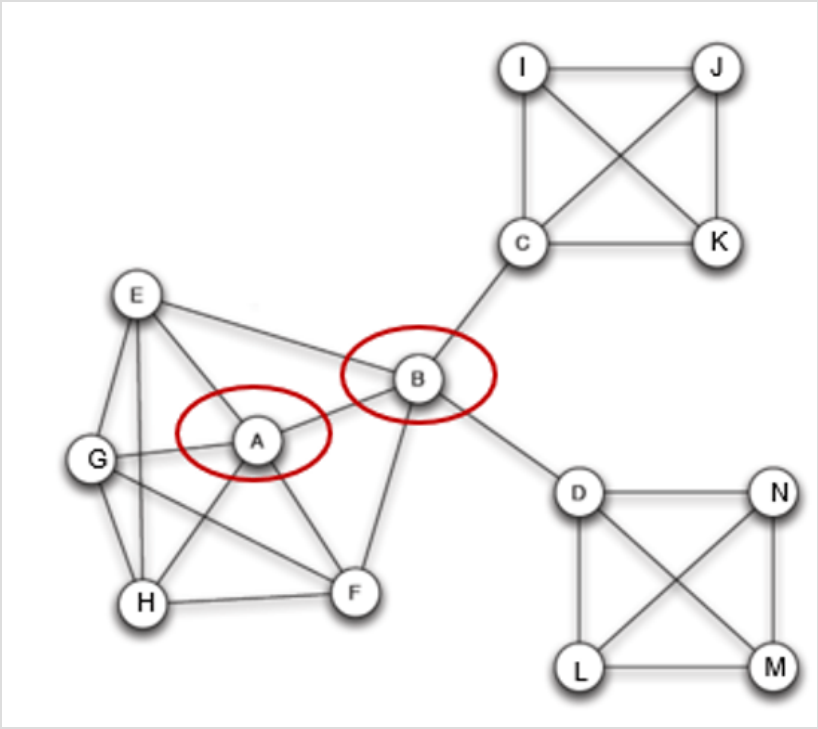
- 每个人都与其他人相连接
- 连接是无向的（我的朋友也把我视为朋友）

How to determine important persons in a social network?

如何确定社交网络中的重要人物?

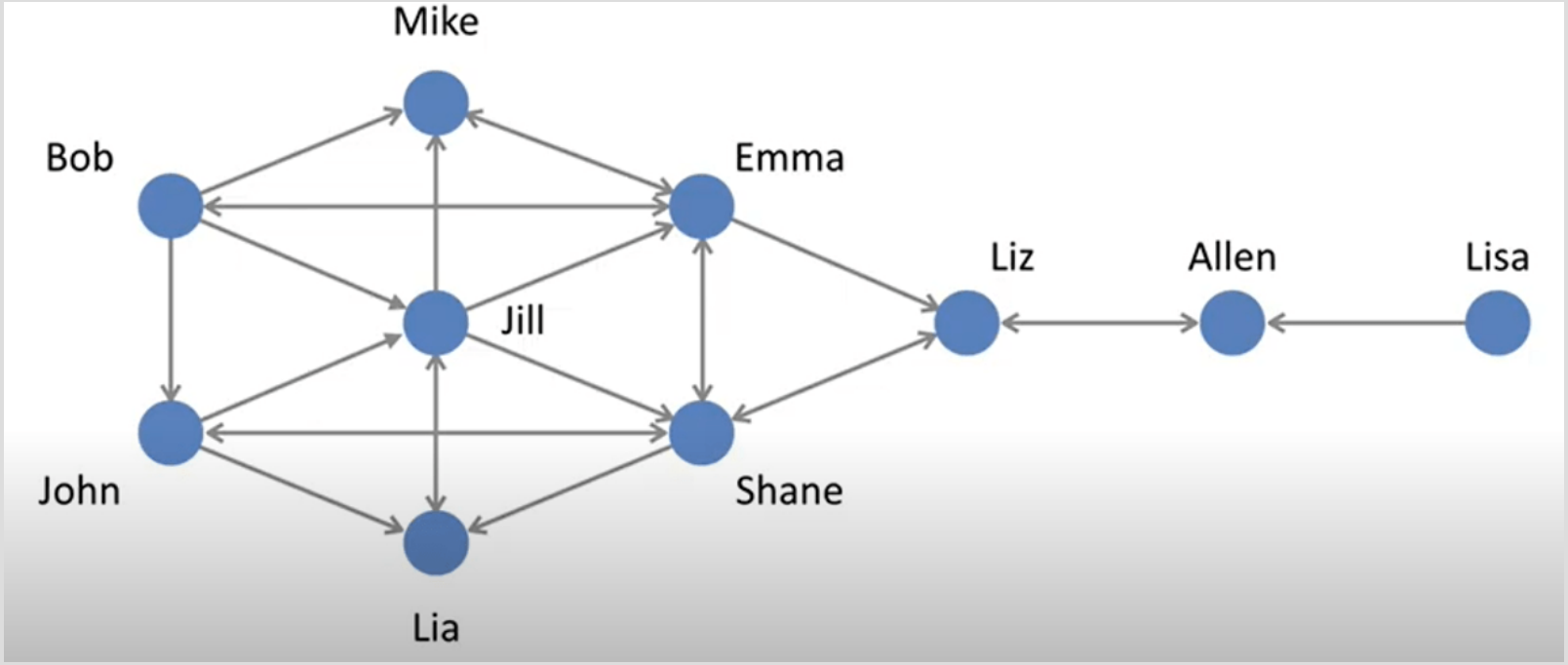
Who is more important? Why?

谁更重要？为什么？



Who is more important? Why?

谁更重要？为什么？



Closeness Centrality

Only applies to a fully connected network (i.e., a path exists between any pair of nodes).

$$\text{closeness centrality}(x) = \frac{N-1}{\sum_y d(x,y)}$$

N: number of nodes in the network

$d(x,y)$: the shortest distance between nodes x and y.

接近中心性

接近中心性仅适用于完全连接的网络(即任意一对节点之间存在路径):

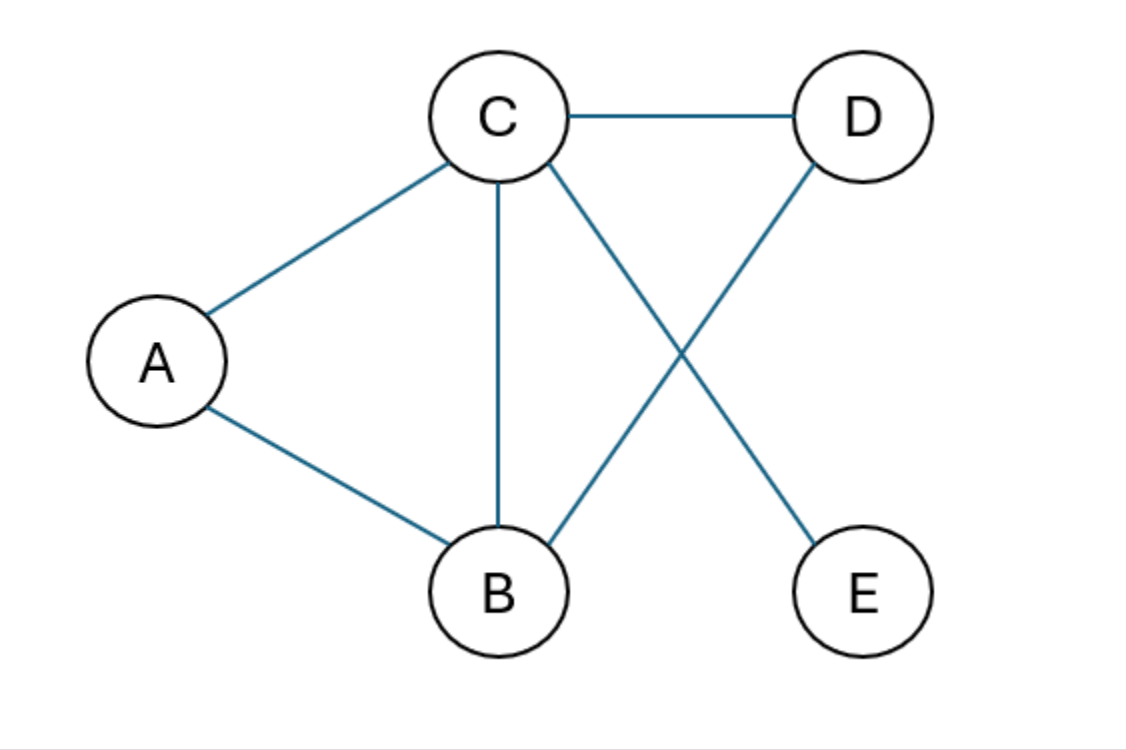
$$\text{closeness centrality}(x) = \frac{N-1}{\sum_y d(x,y)}$$

N: 网络中的节点数量

$d(x,y)$: 节点 x 和 y 之间的最短距离.

What is the closeness centrality of *A*?

A 的接近中心性是多少?



New chat

Generate an undirected graph with 5 nodes, A, B, C, D, E, with the following 6 edges (A, B), (A, C), (B, C), (B, D), (C, D) and (C, E).

Step 1: Shortest path distances from A

- A to A: 0 (distance to itself is 0).
- A to B: 1 (direct edge (A, B)).
- A to C: 1 (direct edge (A, C)).
- A to D: 2 (path $A \to B \to D$ or $A \to C \to D$).
- A to E: 2 (path $A \to C \to E$).

Step 2: Sum of distances

$$\text{Sum of distances} = 0 + 1 + 1 + 2 + 2 = 6$$

Step 3: Closeness centrality formula

$$\text{Closeness Centrality}(A) = \frac{\text{Number of other nodes}}{\text{Sum of distances}} = \frac{4}{6} = \frac{2}{3} \approx 0.6667$$

Answer generated by DeepSeek

Betweenness Centrality

Applies to disconnected networks as well.

$$\text{betweenness centrality}(x) = \sum_{y,z} \frac{\sigma_{yz}(x)}{\sigma_{yz}}$$

σ_{yz} is the total number of shortest paths from y to z.
$\sigma_{yz}(x)$ is the number of shortest paths from y to z that go through x.

中介中心性

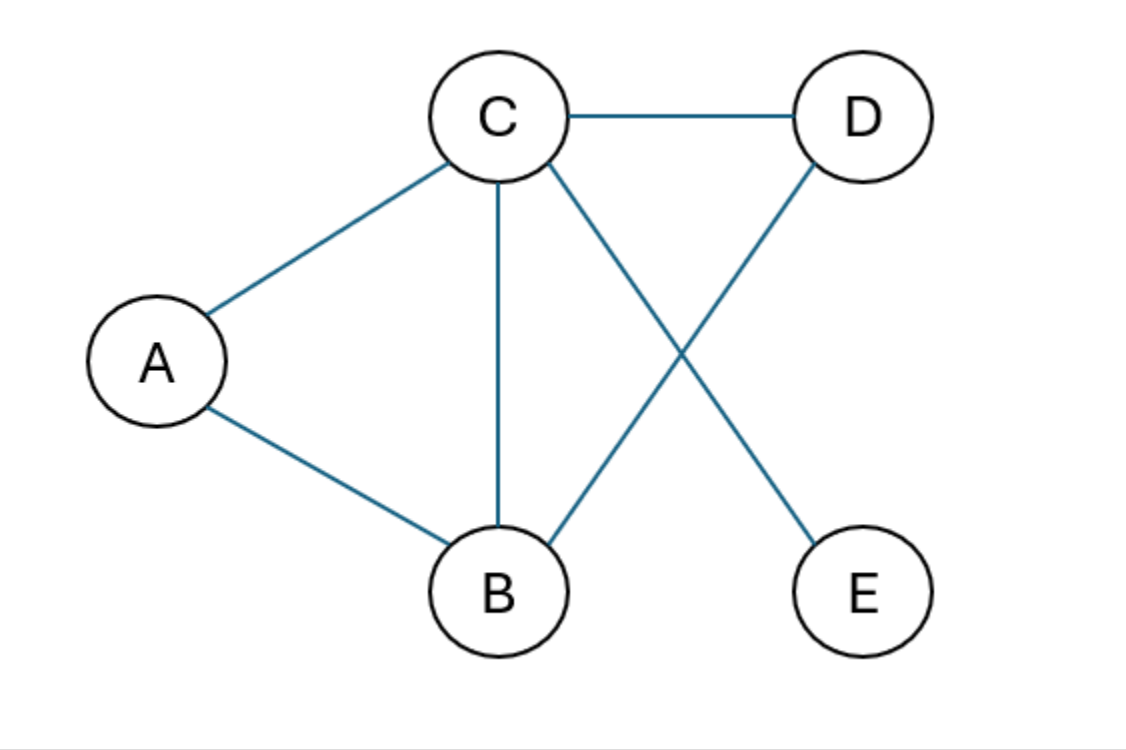
中介中心性同样适用于不连通的网络.

$$\text{betweenness centrality}(x) = \sum_{y,z} \frac{\sigma_{yz}(x)}{\sigma_{yz}}$$

σ_{yz} 是 y 到 z 之间最短路径的条数.
$\sigma_{yz}(x)$ 是 y 到 z 的最短路径中，穿过 x 的最短路径的条数.

What is the betweenness centrality of C?

C 的中介中心性是多少?



Pairs **点对**	**Short paths** **最短路径**	**Shorts path through C** **经过C的最短路径**
(A, B)	1 (A-B)	0
(A, D)	2 (A-B-D; A-C-D)	1 (A-C-D)
(A, E)	1 (A-C-E)	1 (A-C-E)
(B, D)	1 (B-D)	0
(B, E)	1 (B-C-E)	1 (B-C-E)
(D, E)	1 (D-C-E)	1 (D-C-E)

$$\text{betweenness centrality}(C) = \frac{1}{2} + 1 + 1 + 1 = 3.5$$

课后讨论问题：

在日常的推荐中，我们认为熟人(例如同一IP地址或者互为好友)往往具有更多的相似性。如何把这一特点融入到我们的推荐系统模型中更好的做推荐？

请点击这个链接或者扫描二维码回答问题