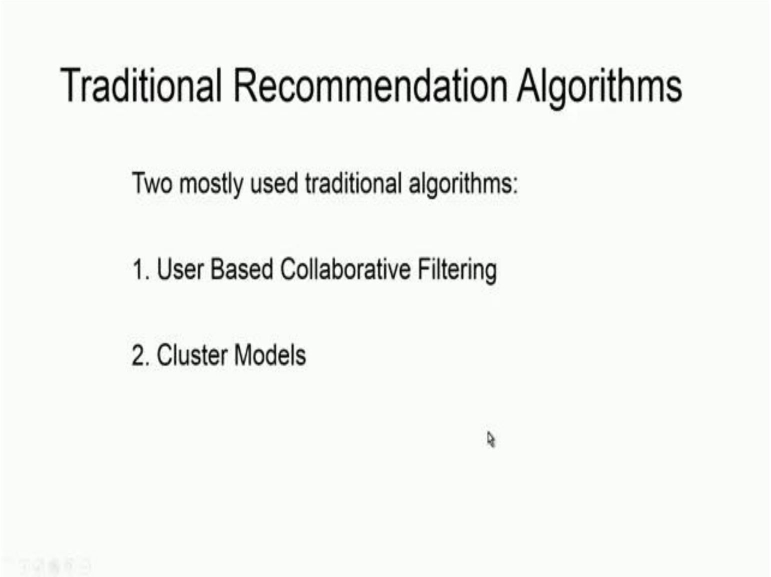
Marketing Analytics
Prof. Swagato Chatterjee
Vinod Gupta School of Management
Indian Institute of Technology, Kharagpur
Lecture 35 - Recommendation Engine and Retail Analytics (Contd.)

Hello everybody, welcome to Marketing Analytics course. This is Dr. Swagato Chatterjee from VGSOM, IIT Kharagpur who is taking this course. We are in week 7, session 2 and we are discussing about recommendation engine and retail analytics. We were discussing specifically about collaborative filtering and I have discussed quite a lot in the last video but I will continue from there.

(Refer Slide Time: 0:38)

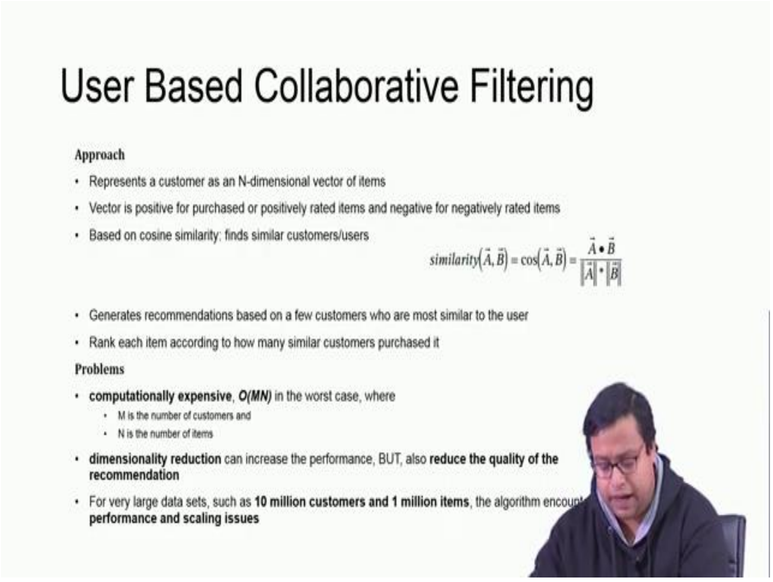


Traditional Recommendation Algorithms

Two mostly used traditional algorithms:

1. User Based Collaborative Filtering

2. Cluster Models



User Based Collaborative Filtering

Approach

- Represents a customer as an N-dimensional vector of items
- Vector is positive for purchased or positively rated items and negative for negatively rated items
- Based on cosine similarity: finds similar customers/users

$$similarity(\vec{A},\vec{B}) = \cos(\vec{A},\vec{B}) = \frac{\vec{A} \bullet \vec{B}}{|\vec{A}| \cdot |\vec{B}|}$$

- Generates recommendations based on a few customers who are most similar to the user
- Rank each item according to how many similar customers purchased it

Problems

- **computationally expensive**, ***O(MN)*** in the worst case, where
    - M is the number of customers and
    - N is the number of items
- **dimensionality reduction** can increase the performance, BUT, also **reduce the quality of the recommendation**
- For very large data sets, such as **10 million customers and 1 million items**, the algorithm encou **performance and scaling issues**

So, here we will talk about user-based and then item-based collaborative filtering. These are the two things we will discuss in details. So the approach is that it is similar. You have to find out a similarity matrix between the customers. So previously I talked about the rating based. In the rating based, we create the similarity based on the correlation of the ratings.

Because ratings are often given in 1 to 5 point scale. We can find out correlations of that. On the other hand, if it is purchased decision based then represents, so approach is represent a customer as a N-dimensional vector of items. So there are items and there are, so what is the dimension of vector? There are n items.

Vector is positive for purchased or positively rated items and negative for negatively purchased or negatively rated items. So the vector is either 1, 0 or 1, -1 or something like that. So 1 means purchase and 0 means not purchased or -1 means not purchased or something like that and then we find out cosine matrix between the customers. So similarity matrix is cosine where I find out a cosine matrix between the customers and then generate recommendations based on a few customers who are most similar. So cosine is the similarity matrix.

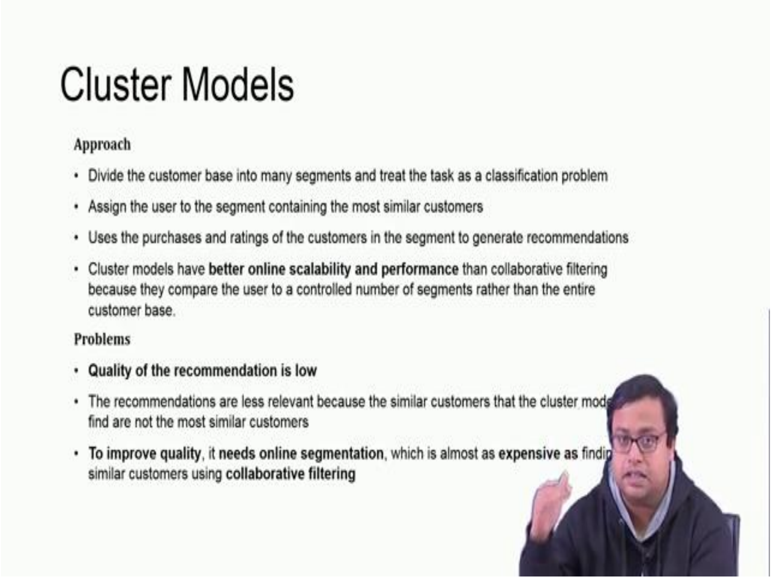
I find out, if I am a customer and there are 10 other customers, whom of these customers' are cosine is smaller? The smaller the cosine, the closer they are to me and if they are closer to me then what is the, sorry, sorry. Cos θ is Cos 0 Sin 0. Cos 0 is small. So smaller the cosine, I would say the closer they are. So generates recommendations based on a few customers who are most similar to the user.

That is something that becomes important and then rank each item according to the how many similar customers have purchased it. So this is something that we will follow in our problem also. So actually it is opposite. Cosine means the smaller is the cosine value. Cosine means the angle.

It actually talks about the angle between the two vectors and smaller is that particular angle the more similar you are and for Cos 0 is 1. Sin 0 is 0. Cos 0 is 1 means specifically, the higher the cosine value the higher is the similarity. Because the theta, the angle is smaller. So that is something that we will try to deal.

Computationally expensive because if there are m number of customers and n number of items, so you have to create a matrix which is m × n sizes and dimensionally reduction is that is why is an important factor, can increase the performance but can also reduce the quality of the recommendation. So that is also a problem. So for very large data sets, such as 10 million customers and 1 million items, the algorithm encounters severe performance and scaling issues and that is why sometimes you have to do batch processing beforehand. You cannot do it in real time. You have to do it beforehand.

(Refer Slide Time: 3:50)



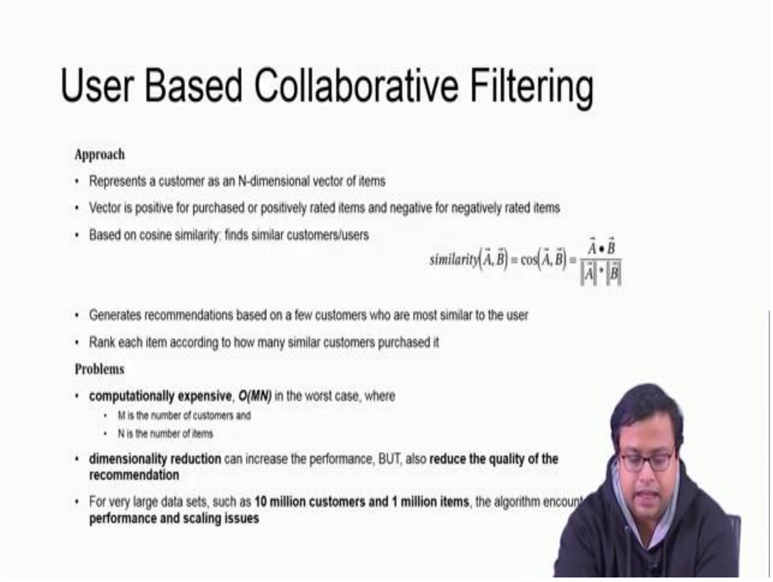
Cluster Models

Approach

- Divide the customer base into many segments and treat the task as a classification problem
- Assign the user to the segment containing the most similar customers
- Uses the purchases and ratings of the customers in the segment to generate recommendations
- Cluster models have **better online scalability and performance** than collaborative filtering because they compare the user to a controlled number of segments rather than the entire customer base.

Problems

- **Quality of the recommendation is low**
- The recommendations are less relevant because the similar customers that the cluster mod find are not the most similar customers
- **To improve quality**, it **needs online segmentation**, which is almost as **expensive as** findi similar customers using **collaborative filtering**



User Based Collaborative Filtering

Approach

- Represents a customer as an N-dimensional vector of items
- Vector is positive for purchased or positively rated items and negative for negatively rated items
- Based on cosine similarity: finds similar customers/users

$$similarity(\vec{A}, \vec{B}) = \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \bullet \vec{B}}{|\vec{A}| \cdot |\vec{B}|}$$

- Generates recommendations based on a few customers who are most similar to the user
- Rank each item according to how many similar customers purchased it

Problems

- **computationally expensive**, ***O(MN)*** in the worst case, where
  - M is the number of customers and
  - N is the number of items
- **dimensionality reduction** can increase the performance, BUT, also **reduce the quality of the recommendation**
- For very large data sets, such as **10 million customers and 1 million items**, the algorithm encoun **performance and scaling issues**

On the other hand, cluster models are different kind of models. The approach is divide the customer base in many segments and treat the task as a classification problem. So first job is to break the customers into various segments. So you do not deal with M number of customers, you deal with small m number of clusters.

Now small m is lower than capital M. So within the cluster I will not focus on the variability of the persons. Assign the user to the segment containing the most similar customers. Fair enough. So it is like, like probably K nearest neighbor model.

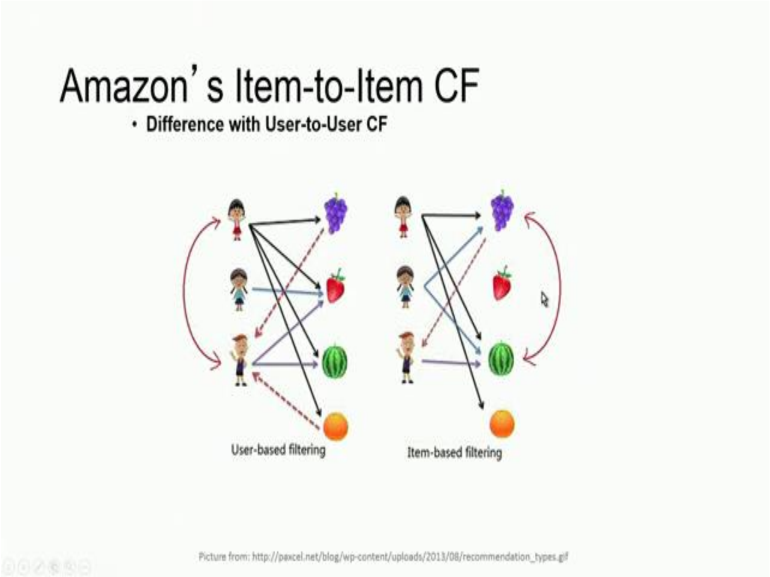
K means clustering model. So if you have clusters and then you find a new cluster, whichever cluster is most closest to this guy, fit them to that cluster. And uses the purchase and ratings of the customers in the segment to generate recommendations. Now when I create the recommendation calculations, I will not focus on other clusters customer, I will focus on only those customer who are in the same segment and then we will do the calculations. Cluster models have better online scalability and performance obviously than collaborative filtering because they compare the user to a controlled number of segments.

What is the problem? As I told the quality is an issue. Reduce the quality of the recommendation when you do dimensionality reduction. So here this is also one type of dimensionality reduction. But here you are not reducing the dimensionality in terms of the number of items.

You are reducing it in terms of number of users. But that also reduces the cost, reduces the quality of the recommendation that you are giving. And the recommendations are less relevant because the similar customers that the clusters find are not the similar, not the most similar customers. So it might be the case that the similar customer that the cluster models find are not actually very close to each other because the method that this clustering method using to create the clusters are not the purchased data. So that is why the similarity might be different.

To improve quality, it needs online segmentation, which is almost as expensive as finding similar customers using collaborative filtering. So in other words, frankly speaking, this is more common than this.

(Refer Slide Time: 6:08)



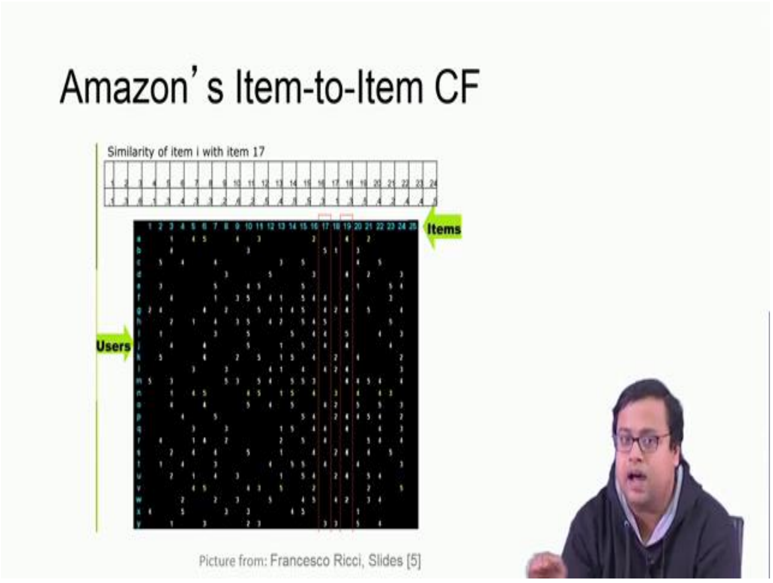
Amazon's Item-to-Item CF

- Difference with User-to-User CF

Picture from: http://paxcel.net/blog/wp-content/uploads/2013/08/recommendation_types.gif

What do you do in a collaborative filtering? In a user to user collaborative filtering or item to item collaborative filtering, there are 2 types of collaborative filtering. In one case you find out the similarity between the users and in other case similarity between the items but in both the cases you are recommending some products to some users based on those matching. So we will try to solve both of them in the next video.

(Refer Slide Time: 6:31)



Amazon's Item-to-Item CF

Picture from: Francesco Ricci, Slides [5]

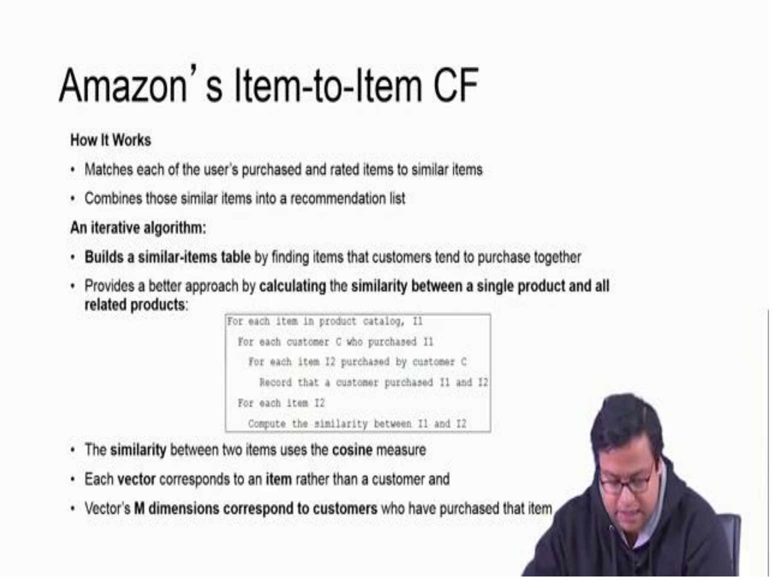
So what we do? Let us say these are 17 items. So similarity of item i with item 17. So let us say there are 25 items in total and a, b, c, d, x, y, z are basically users and these 25 items are there. Fair enough. And then I try to find out let us say item number 17 with, so here it is actually 25 is missing. So these values that you are seeing here all these values are basically the number of purchases that has been made. So, for now 17, 0, 5, 000, 44444 etc etc. So, 17 this particular product has been bought by these people.

Now how will I know that two items are similar? If two items are bought by similar type of people then they are similar. So whoever has seen 17, if they have also seen 19 then 17 and 19 are similar products. So, that is how if I do a column wise calculation, so column wise correlation matrix, I will get the similarity or cosine matrix let us say. I will get the similarity of item 1 with item 2 with item 3 and so on.

On the other hand, same thing if I do the row wise, I will find out how customer A and customer B are similar. See, customer A has seen, both have not seen 1 and 2. Customer A has seen 3. Customer B has seen 3 four times. 4 both of them have not seen.

Up to 9 customer B has seen nothing but 4, 5 and 5, 6 and 9 customer A has seen. So, that is a difference. So based on the purchased rate of customer A and purchase rate of customer B, I can find out how customer A and customer B are similar to each other or dissimilar to each other. So as simple as that if you do column wise in this particular matrix, you do column wise, you get item to item collaborative filtering. If you do row wise, you will get user to user collaborative filtering.

(Refer Slide Time: 8:44)



Amazon's Item-to-Item CF

How It Works

- Matches each of the user's purchased and rated items to similar items
- Combines those similar items into a recommendation list

An iterative algorithm:

- **Builds a similar-items table** by finding items that customers tend to purchase together
- Provides a better approach by **calculating** the **similarity between a single product and all related products**:

```
For each item in product catalog, I1
  For each customer C who purchased I1
    For each item I2 purchased by customer C
      Record that a customer purchased I1 and I2
  For each item I2
    Compute the similarity between I1 and I2
```

- The **similarity** between two items uses the **cosine** measure
- Each **vector** corresponds to an **item** rather than a customer and
- Vector's **M dimensions correspond to customers** who have purchased that item

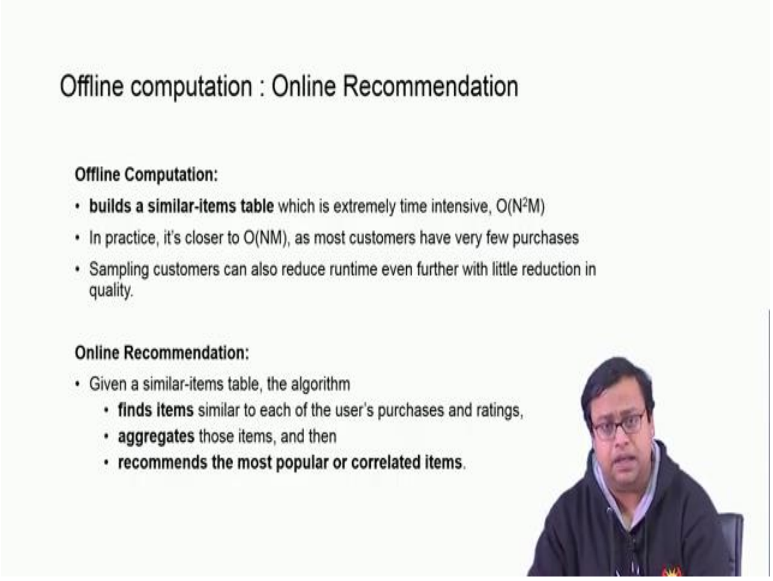
So, item to item how it works? Matches each of the user's purchased and rated items to similar items. So these are actually ratings. So each of the items to the similar items combines those similar items to a recommendation list. Then what is an iterative algorithm? Build a similar items table which will create and provides a better approach by calculating the similarity between a single product to all related products.

The similarity between two items use the cosine measure. The same thing, each vector corresponding to item rather than a customer. So it corresponds to an item and vector of M dimensions corresponding to customers who have purchased that item.

So see, the advantage is that the number of items in e-commerce firm is generally lesser than the number of customers. In a already established e-commerce firm less number of products are there, more number of customers are there. Just give an idea, so how many products you have purchased in an e-commerce in Amazon or Flipkart in the last 1 year let us say? 10. So when a person buying 10 items, 15 items, so that is saying that okay, the number of items are lower, number of customers are probably, number of items are lower, number of customers are higher because same item have been bought by multiple customers.

So generally, that is what is being seen and because that number of products are lower, it is better to do item to item collaborative filtering because that takes less time.

(Refer Slide Time: 10:18)



Offline computation : Online Recommendation

Offline Computation:

- **builds a similar-items table** which is extremely time intensive, O(N^2M)
- In practice, it's closer to O(NM), as most customers have very few purchases
- Sampling customers can also reduce runtime even further with little reduction in quality.

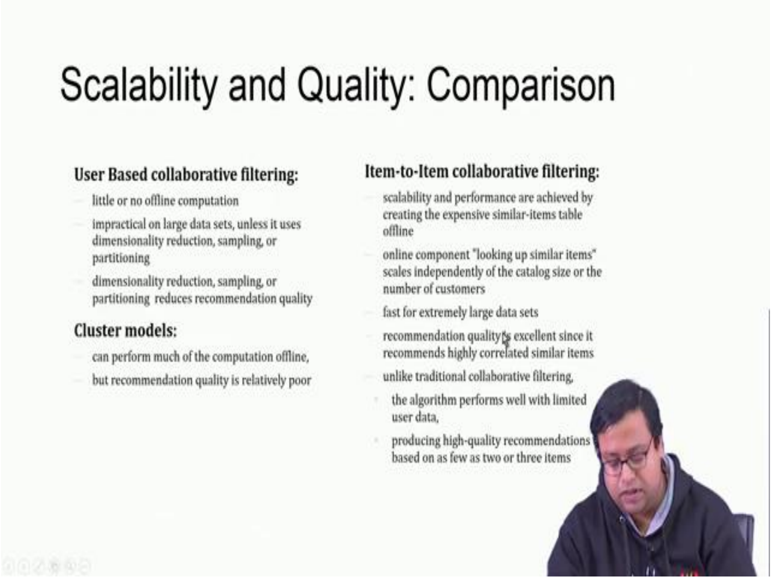
Online Recommendation:

- Given a similar-items table, the algorithm
    - **finds items** similar to each of the user's purchases and ratings,
    - **aggregates** those items, and then
    - **recommends the most popular or correlated items**.

Offline computation has to be done because builds a similar item table that calculation will trigger because lots of time it is extremely time intensive. In practice, it is closer to OMN that means as most customers have very few purchases and sampling customers can also reduce runtime even further with a little reduction of similarity. So that is something.

Online recommendation give a similar item tables, the algorithm finds items similar to each of the user’s purchased and ratings. Aggregate those items and then recommends the most popular correlated items. So these are all the texts. Let us do something hands on and then you will be able to understand what we are trying to talk about.

(Refer Slide Time: 11:02)



Scalability and Quality: Comparison

User Based collaborative filtering:

- little or no offline computation
- impractical on large data sets, unless it uses dimensionality reduction, sampling, or partitioning
- dimensionality reduction, sampling, or partitioning reduces recommendation quality

Cluster models:

- can perform much of the computation offline,
- but recommendation quality is relatively poor

Item-to-Item collaborative filtering:

- scalability and performance are achieved by creating the expensive similar-items table offline
- online component "looking up similar items" scales independently of the catalog size or the number of customers
- fast for extremely large data sets
- recommendation quality is excellent since it recommends highly correlated similar items
- unlike traditional collaborative filtering,
  - the algorithm performs well with limited user data,
  - producing high-quality recommendations based on as few as two or three items

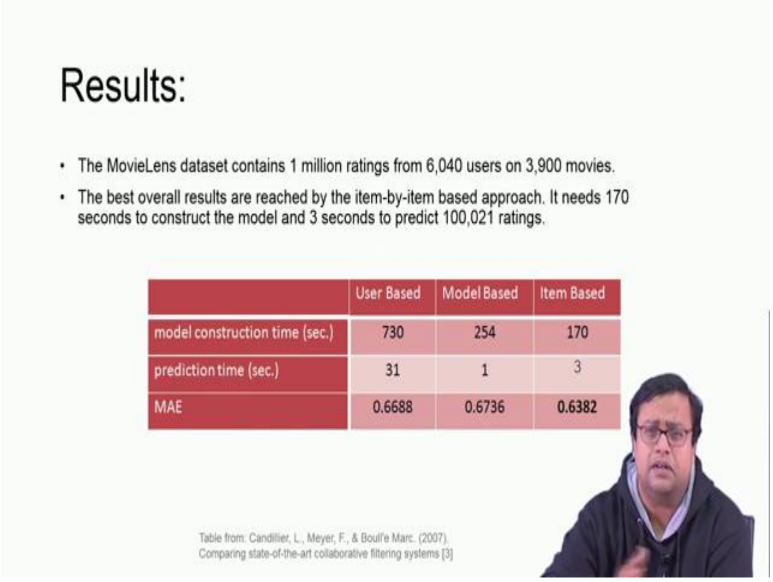
Now if I compare these 3 basic measures in terms of the scalability and quality, basically these are the two issues whether it is time taking, whether it is the quality that you are getting. So user based collaborative filtering requires no offline computation, impractical or large data sets because it takes lots of time and dimensionally reduction, sampling and partitioning reduces recommendation quality.

On the other hand, cluster models can perform much on a computation offline but recommended quality is relatively poor. So that is a problem. So these two cases it works well but for the moment takes time. The moment you will try to reduce the time taking, its quality gets reduced and in general, quality is low.

On the other hand, item to item collaborative filtering, scalability and performance are achieved by creating the expensive similar items table offline. So batch processing is done beforehand. An online component looking up similar item scales independently of the catalog size or the number of customers. So it does not depend on the number of customers. It only depends on the catalog size.

So catalog, so then it is okay. First for extremely large data sets it works. Recommendation quality is excellent since it recommends high correlated similar items. So basically it is saying that this item to item collaborative filtering works better than these two. We will solve all these three but we will try to focus on this one more.

(Refer Slide Time: 12:30)



Results:

- The MovieLens dataset contains 1 million ratings from 6,040 users on 3,900 movies.
- The best overall results are reached by the item-by-item based approach. It needs 170 seconds to construct the model and 3 seconds to predict 100,021 ratings.

	User Based	Model Based	Item Based
model construction time (sec.)	730	254	170
prediction time (sec.)	31	1	3
MAE	0.6688	0.6736	**0.6382**

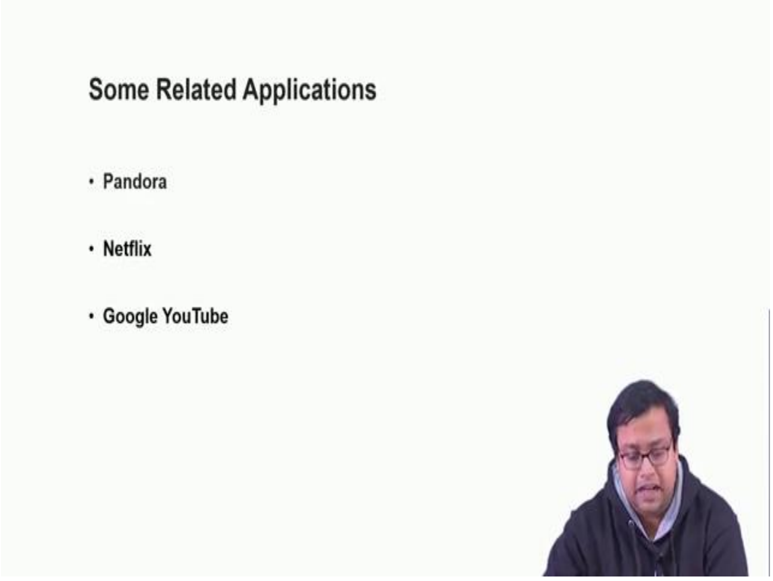
Table from: Candillier, L., Meyer, F., & Boull'e Marc. (2007). Comparing state-of-the-art collaborative filtering systems [3]

So some of the results the MovieLens dataset, some people have tried these 3 methods. The MovieLens and the reference is given below. The MovieLens dataset 1 million ratings of 6040 users and 3090 movies. So number of movies are lower than number of users. The best overall results are reached by item to item collaborative filtering. You see that item to item collaborative filtering to 170 second. Much much lower than the other one and a mean average, mean absolute radar means whatever the recommended people actually bought it or not.

The radar is coming up to 63 %

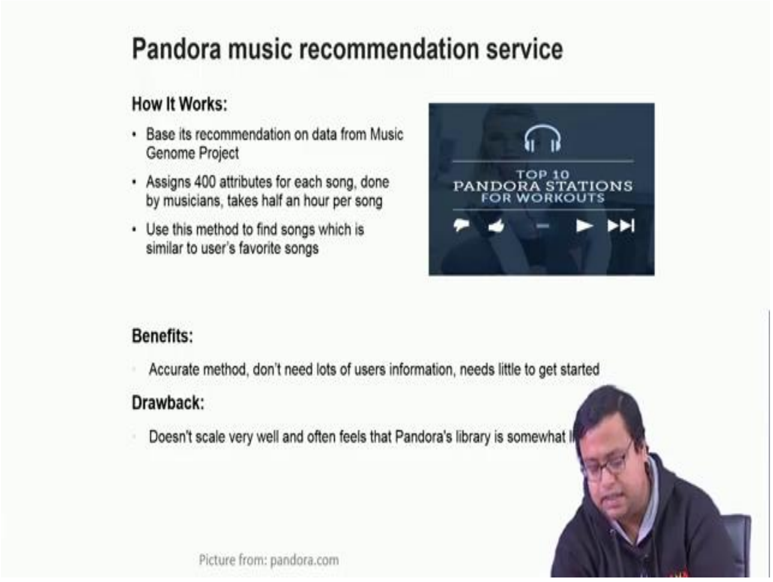
. So it is not 63, 0.6382. In other case, it is 0.66, 0.67 so comparable. And prediction time taken is 3 second. So item to item giving similar result but with much much lesser time.

(Refer Slide Time: 13:22)



Some Related Applications

- Pandora
- Netflix
- Google YouTube



Pandora music recommendation service

How It Works:

- Base its recommendation on data from Music Genome Project
- Assigns 400 attributes for each song, done by musicians, takes half an hour per song
- Use this method to find songs which is similar to user's favorite songs

Benefits:

Accurate method, don't need lots of users information, needs little to get started

Drawback:

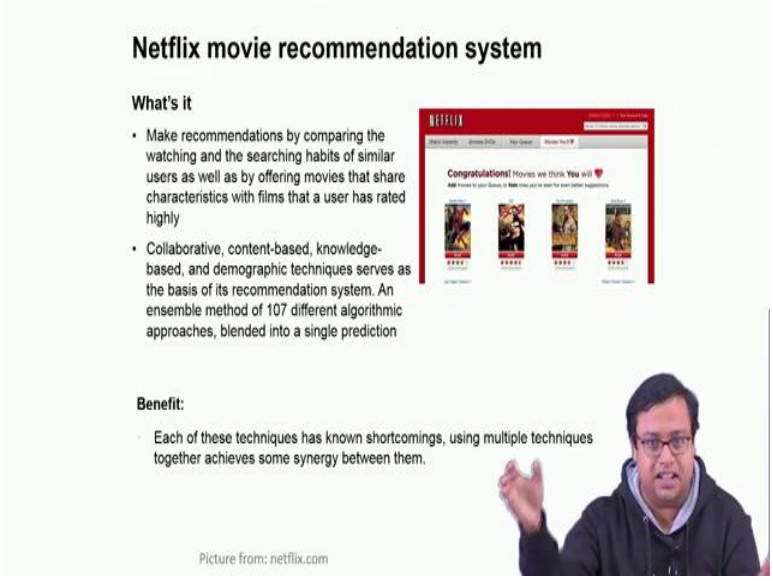
Doesn't scale very well and often feels that Pandora's library is somewhat l

Picture from: pandora.com

Some related applications can be Pandora, can be Netflix, can be Google YouTube. For example, for Pandora, Pandora is based its recommendation on data for Music Genome Project. Assigns 400 attributes for each song. Each song they find out 400 attributes. Done by musicians. Take half an hour per song and use this method to find songs which is similar to user's favorite songs. What is the method, benefit? Accurate method do not need lots of user information.

So because it is item based music and musician are matched. Do not scale very well and often feels that Pandora's library is somewhat limited. So that is one problem.

(Refer Slide Time: 14:03)



Netflix movie recommendation system

What's it

- Make recommendations by comparing the watching and the searching habits of similar users as well as by offering movies that share characteristics with films that a user has rated highly
- Collaborative, content-based, knowledge-based, and demographic techniques serves as the basis of its recommendation system. An ensemble method of 107 different algorithmic approaches, blended into a single prediction

Benefit:

- Each of these techniques has known shortcomings, using multiple techniques together achieves some synergy between them.

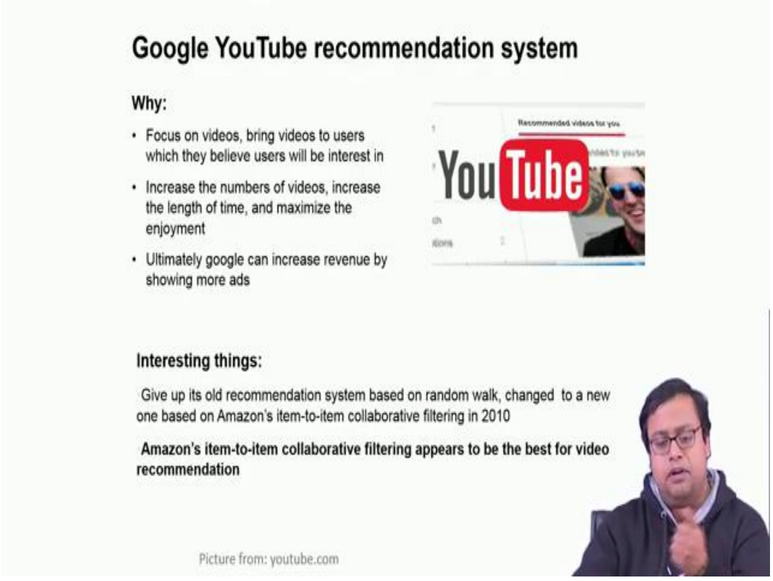
Picture from: netflix.com

Netflix's movie recommendation based on the ratings. Make recommendations by comparing the watching and the searching habits. Sorry, searching habits of similar users as well as by offering movies that share characteristics with films that a user has rated. So both the purchase behavior of the customer and the content, so create a hybrid model. Collaborative, content based, and knowledge based and demographic techniques serve as the basis of the recommendation system. So collaborative filtering is used, content is used and also the demographic of the customers is used.

So it is ensemble method. It is a user based, item based and content based then create a recommendation engine with all these three and then match them together and create an ensemble method of 107 different algorithm approaches. What is the benefit? So these are large scale implementations.

We will do small small examples. Each of these techniques have some short comings, use multiple techniques together to overcome, so one shortcoming of one technique is overcome by the advantage of another technique. So that is how the ensemble methods work.

(Refer Slide Time: 15:08)



Google YouTube recommendation system

Why:

- Focus on videos, bring videos to users which they believe users will be interest in
- Increase the numbers of videos, increase the length of time, and maximize the enjoyment
- Ultimately google can increase revenue by showing more ads

Interesting things:

Give up its old recommendation system based on random walk, changed to a new one based on Amazon's item-to-item collaborative filtering in 2010

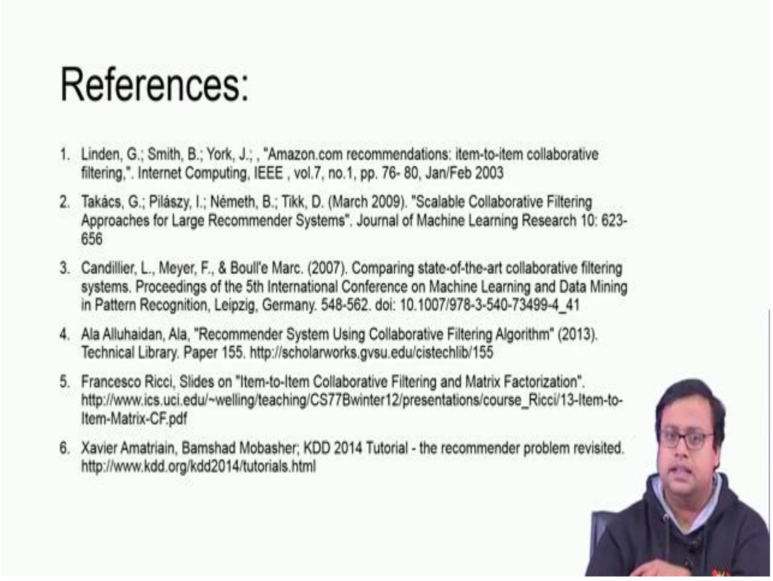
Amazon's item-to-item collaborative filtering appears to be the best for video recommendation

Picture from: youtube.com

Google YouTube recommendation system how it works? It focus on videos. Brings videos to users which they believe users will be interested in. Increase the number of videos, increase the length of time and maximize the enjoyment and this is their motto. Ultimately, Google can increase revenue by showing more ads. So more the engagement, the more ads they can show and the more revenue they can generate. What is it?

Give up the old recommendation system based on random walk. Change to a new one based on Amazon’s item to item collaborative filtering. So old one was some random one and this one is based on collaborative filtering. In 2010, they did this change. Amazon’s item to item collaborative filtering appears to be the best for video recommendation.

(Refer Slide Time: 15:52)



References:

1. Linden, G.; Smith, B.; York, J.; , "Amazon.com recommendations: item-to-item collaborative filtering,". Internet Computing, IEEE , vol.7, no.1, pp. 76- 80, Jan/Feb 2003
2. Takács, G.; Pilászy, I.; Németh, B.; Tikk, D. (March 2009). "Scalable Collaborative Filtering Approaches for Large Recommender Systems". Journal of Machine Learning Research 10: 623-656
3. Candillier, L., Meyer, F., & Boull'e Marc. (2007). Comparing state-of-the-art collaborative filtering systems. Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition, Leipzig, Germany. 548-562. doi: 10.1007/978-3-540-73499-4_41
4. Ala Alluhaidan, Ala, "Recommender System Using Collaborative Filtering Algorithm" (2013). Technical Library. Paper 155. http://scholarworks.gvsu.edu/cistechlib/155
5. Francesco Ricci, Slides on "Item-to-Item Collaborative Filtering and Matrix Factorization". http://www.ics.uci.edu/~welling/teaching/CS77Bwinter12/presentations/course_Ricci/13-Item-to-Item-Matrix-CF.pdf
6. Xavier Amatriain, Bamshad Mobasher; KDD 2014 Tutorial - the recommender problem revisited. http://www.kdd.org/kdd2014/tutorials.html

So this is somehow the background. So we have learned about basically some of the references. We have learned the primary features about item to item, user to use collaborative filtering, content based method and then the clustering method, cluster based recommendation engine. So these are the 4 things that we will try to use in smaller problems in a hands on basis in the next videos. Thank you for being with me with this video. I will come back with the next video in terms of collaborative filtering. First, we will do item to item. Thank you very much.