JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Lecture 4

BU.330.740 Large Scale Computing on the Cloud

Minghong Xu, PhD.
Associate Professor

Homework 1 Review

- Use complete sentences to describe
  - No points deducted this time

- Doubleton
  - Mapper key: two items in format (item i, item j)
  - Mapper value: 1
  - Reducer key: still item pair
  - Reducer value: summation/count/support
  - Reducer operation: summation

Homework 1 Review (Cont.)

- Plagiarism
  - Mapper 1 key: sentenceID; value: articleID
  - Reducer 1 key: sentenceID; value: list of articles
  - Mapper 2 key: article pair; value: 1
  - Reducer 2 key: article pair; value: summation

Reflections

- Spark: plugin for engines to speed up processing using distributed computing
- pySpark
  - RDD, actions vs transformations, DataFrame, ML
- Data pipeline
- ETL vs ELT
  - ETL for structured and operational data, usually loaded into data warehouse
  - ELT for big data and advanced analytics, usually loaded into data lake
  - Can be used together in hybrid data pipeline

Today’s Agenda

- Scaled machine learning pipeline and AWS SageMaker
- Recommendation systems
- Lab3: movie recommender in AWS SageMaker

Business Insider

Meta speeds up its hiring process for machine-learning engineers as it cuts thousands of 'low performers'

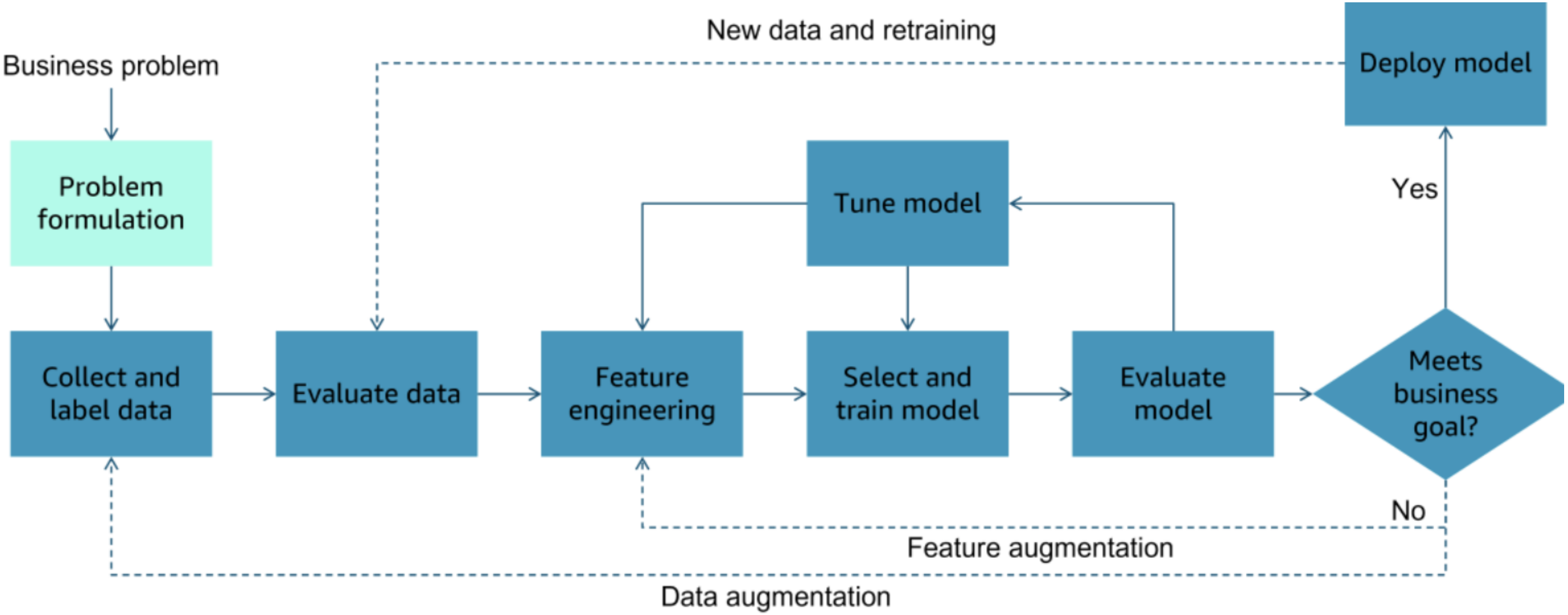
Meta is ramping up hiring for machine-learning engineers while slashing thousands of jobs. Last month Meta said it would eliminate about 5%...

23 hours ago

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Scaled Machine Learning Pipeline

Machine Learning Pipeline



Why Scaling

- Big Data
- Distributed processing
- Real-time needs

- A system designed to automate, manage, and optimize the process of building, deploying, and maintaining machine learning models at scale

Scaled Pipeline

- Data ingestion: can handle large volumes of structured, semi-structured, or unstructured data
  - Tools: Apache Kafka, AWS Kinesis
  - Scalable Storage Solutions: HDFS, S3, Google Cloud Storage
- Data preprocessing: distribute preprocessing tasks across multiple machines
  - Tools: Apache Spark

Components (Cont.)

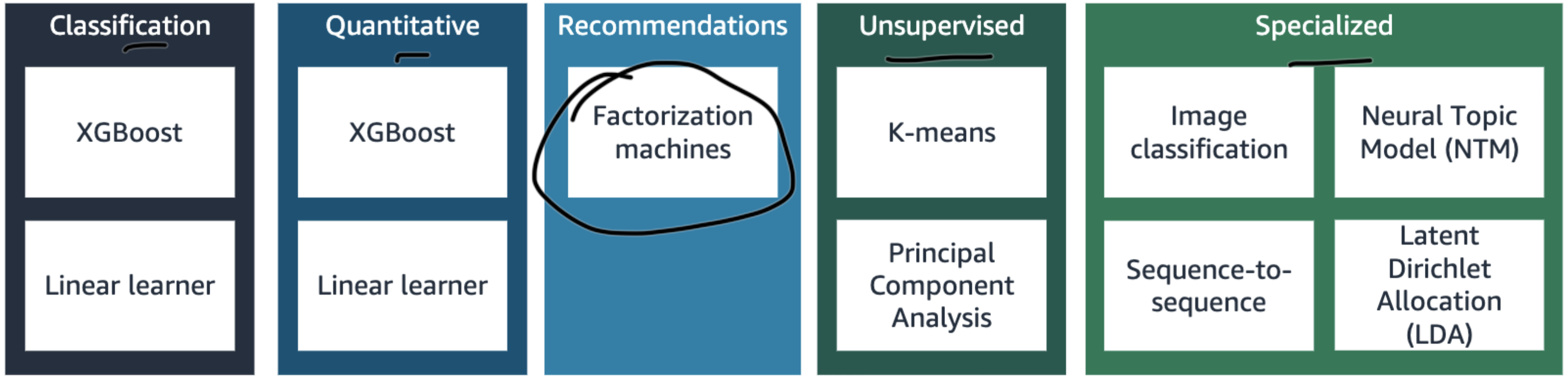
- Model training: use distributed computing frameworks to parallelize training

  Meta Amazon Google MSFT.

  - PyTorch Distributed, SageMaker, Vertex AI, Azure ML
- Model deployment: can handle high throughput and low latency requirements at scale
  - Tools: TensorFlow Serving, NVIDIA Triton, Kubeflow Pipelines
  - Serverless: AWS Lambda, Google Cloud Functions

Amazon SageMaker

- Amazon SageMaker provides ML algorithms that are optimized for speed, scale, and accuracy
- Built-in algorithms:



AdaBoost vs Gradient Boost

Linear regression vs Logistic regression

Supported Algorithms

- Supported frameworks
  - TensorFlow
  - PyTorch
  - scikit-learn
  - SparkML

DL

ML

- Marketplace algorithms: AWS Marketplace lists ready-to-use algorithms and models developed by third-party
  - https://aws.amazon.com/marketplace

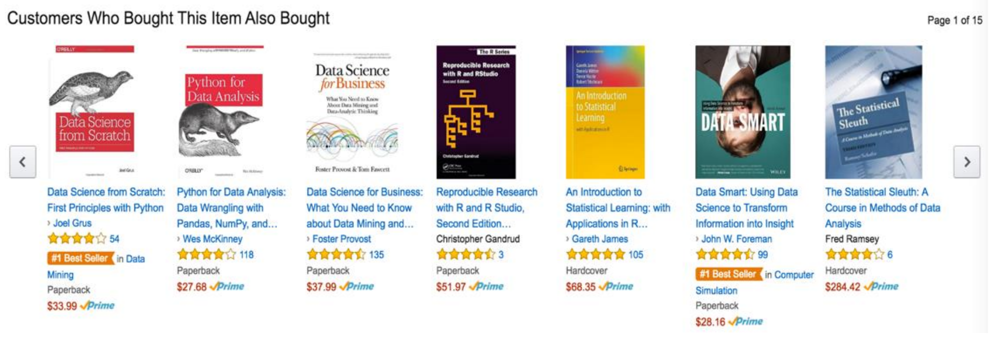
Other Features

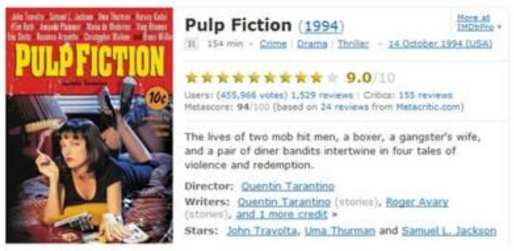
- Automated hyperparameter tuning
  - https://sagemaker.readthedocs.io/en/stable/api/training/tuner.html
- Autopilot: automatically find a good model
  - You: create a job, supply test, training, and target
  - Autopilot: analyze data, select appropriate features, and train and tune models
- Scaled deployment
  - Host the model after trained, handle requests via internet
  - Hosting model is expensive

Recommender System

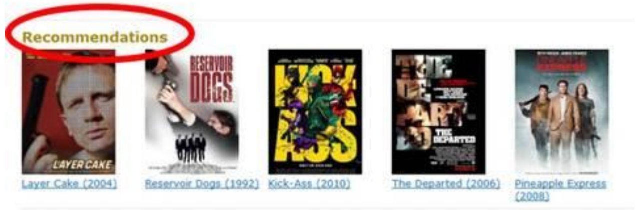
Age of Discovery

- *“The Web, they say, is leaving the era of search and entering one of discovery. What's the difference? Search is what you do when you're looking for something. Discovery is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you.”* (CNN Money)





More information about the movie.....



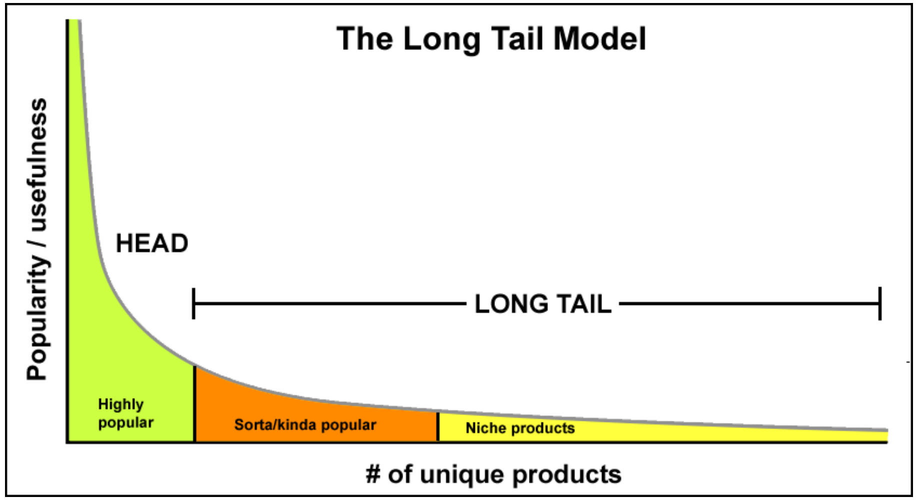
Recommender and the Long Tail

- Traditional brick-and-mortar store *v.s.* on-line store
  - **Netflix**: While Netflix carries many popular shows and movies, it also carries just as many (if not, more) less popular titles. The less popular titles contribute to the overall watch time and attract niche visitors
  - **Amazon**: Amazon sells/lists 600 million products, and the number of low demand products is equal to or more than the high demand products (https://amzscout.net/blog/amazon-statistics/)

- The distinction, called *long tail phenomenon* makes recommenders necessary

Long Tail Marketing

- Long tail marketing: strategy of targeting a large number of niche markets



sales diversity ensures revenue in long term.

- Recommendation Engine: aim to address the product selection problem, by using data on purchases, product ratings, and user profiles to predict which products are best suited to a particular user

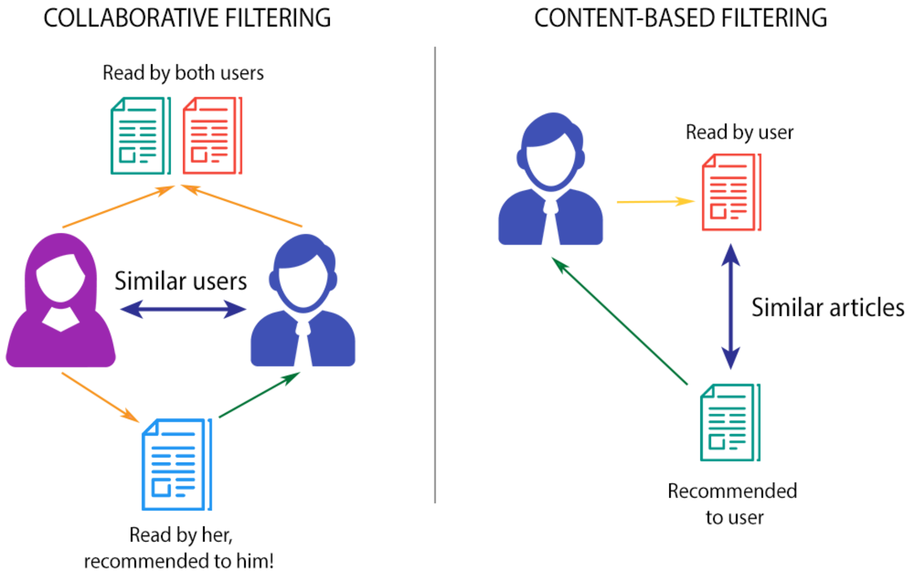
Recommender System

① How can we find the customers interest? like which products they are interested to buy.

- *Predict* users' responses to options
- Two classes of entities: *users* and *items*
- Users have preferences towards items. *How can we know it?*
  - Explicit Ratings
    - Explicitly expressed, such as ratings with stars, or reviews (sentiment analysis applied)
    - Willingness of the users required
  - Implicit Ratings
    - Interactions with items are interpreted as expressions of preference, such as purchasing a book, reading an article
    - Interactions must be detectable

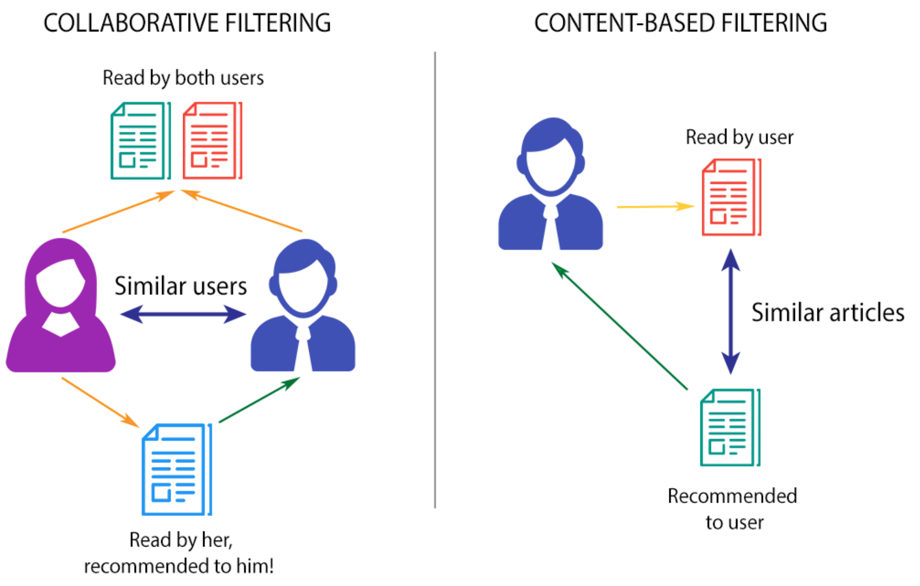
Two Basic Approaches

- Content-based systems (CB)
- Utilize a set of discrete characteristics of an item
- Recommend additional items with similar properties



Two Basic Approaches (Cont.)

- Collaborative filtering-based systems (CF)
- More common approach
- Collect human judgements or ratings and match together people who share the same information needs or the same tastes
- Recommend products bought or liked by other consumers who are similar in tastes



Process of CB

- Given the user's purchased and rated items, construct a search query to find other popular items, similar items
  - E.g., same author/artist/director, similar keywords/subjects
- Based on contents rather than other users' opinions/interactions
- Common for recommending text-based products
  - Web pages, news messages
- Items to recommend are described by features, such as keywords
  - Discover features of documents via text mining
  - Obtain item features from tags, topic modeling to generate tags

CF: Utility Matrix

- User-item pair, a value represents the degree of preference of that user for that item
- Task is to predict unknown/missing entries by finding patterns in the known entries
- Not necessary to predict every blank entry
- Only discover some entries in each row that are likely to be high
- Real utility matrix can be **huge** and **sparse**

A Sample User-Item-Matrix

	The Matrix	Alien	Inception
Alice	5	1	4
Bob	?	2	5
Peter	4	3	2

	SHERLOCK	HOUSE of CARDS	AVENGERS	ARRESTED DEVELOPMENT	Breaking Bad	WALKING DEAD	sim(u,v)
	2		2	4	5		NA
	5		4			1	
			5		2		
		1		5		4	
			4			2	
	4	5		1			NA

Nearest-Neighbor Approach

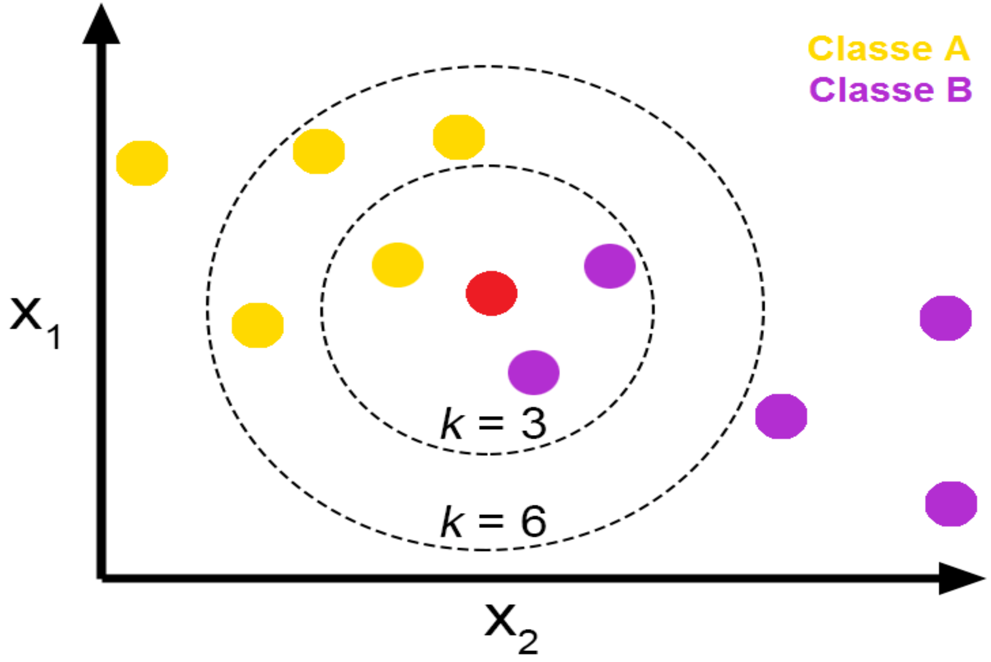
- Nearest-Neighbor algorithm: Users are similar if their vectors are close
- *Any issue?*
- Measure similarity of every pair of customers (a&b, a&c, b&c, etc.), can be computationally expensive

A Sample User-Item-Matrix



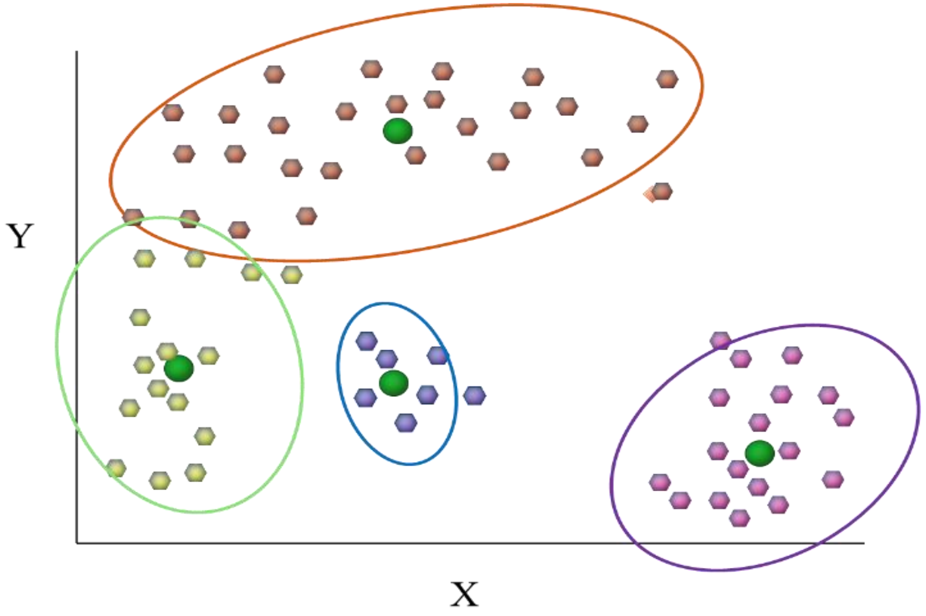
	The Matrix	Alien	Inception
Alice	5	1	4
Bob	?	2	5
Peter	4	3	2

Alice's vector: [5,1,4]



Clustering Approach

- Identify groups of consumers who appear to have similar preferences
- Computationally less expensive than KNN. *But issues?*
- May hurt accuracy while dividing the population into clusters
  - Some data points in the same cluster may still be very different



Item-item Instead of User-user

- No more matching the user to similar customers
- Build a similar items table by shared customers
- Invented and used by Amazon in 1998

A Sample User-Item-Matrix

	The Matrix 	Alien 	Inception 
Alice	5	1	4
Bob	?	2	5
Peter	4	3	2

- *Why?*

Inception's vector: [4,5,2]

- The **item-item scheme is fairly static**, which can be precomputed offline to improve the online performance

Similarity Measure

- *How to measure similarity between items?*
  - Correlation between ratings, or
  - Cosine of those rating vectors
- Correlation (Review)

Given a series of n measurements of the pair (X_i, Y_i) indexed by $i = 1, \ldots, n$, the *sample correlation coefficient* can be used to estimate the population Pearson correlation $\rho_{X,Y}$ between X and Y. The sample correlation coefficient is defined as

$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{n s'_x s'_y}$$

$$= \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}}.$$

- Implemented in Lab3 Extension using Apache Pig

Cases

- Amazon
- [Two Decades of Recommender Systems at Amazon.com]

- *Major content-based approach business?*

- TikTok
- [TikTok's AI Strategy]

Discussion Time: CB vs CF

- Content Based vs Collaborative filtering Based
- *Advantages and disadvantages of each?*

Pros and Cons

- CB requires feature information
- CB only make recommendations based on existing interests
- CF requires data of other users
- “Cold-start” issue

- **Features vs Tastes**
- Can be combined in hybrid systems

Value of Recommenders

- Netflix: 2/3 of the movies watched are recommended
- Google News: recommendations generate 38% more clickthrough
- Amazon: 35% sales from recommendations

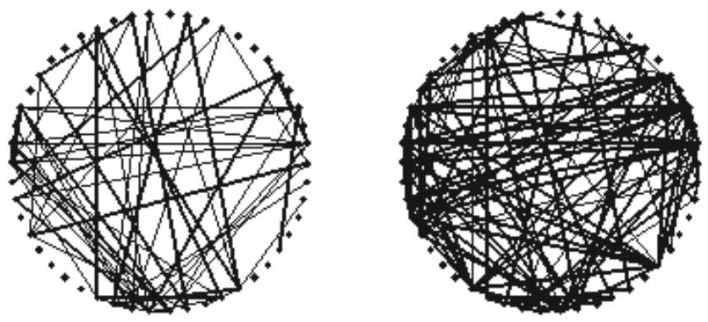
- Utilized for a variety of items: movies, music, books, jokes, news, wines, restaurants, garments, research papers, experts and research collaborators, search queries, social tags, financial services, life insurance plans, nursing care plans, matchmaking, social media contents, …

Recommender and Long Tail

- Recommendation networks and the long tail of electronic commerce (Oestreicher-Singer and Sundararajan 2012 *MIS quarterly*: 65-83)
  - Revenue distributions of books in over 200 categories on Amazon
  - Categories whose products are influenced more by the recommendation network have significantly flatter demand and revenue distributions
  - Average 50% increase in revenue of least popular 20%
  - Average 15% reduction in revenue of most popular 20%

Recommender and Sales Diversity

- Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity (Fleder and Hosanagar 2009 *Management science* 55.5: 697-712)
  - Recommenders can push each person to new products, but they often push users toward the same new products
  - Individual-level diversity to increase
  - Aggregate diversity to decrease

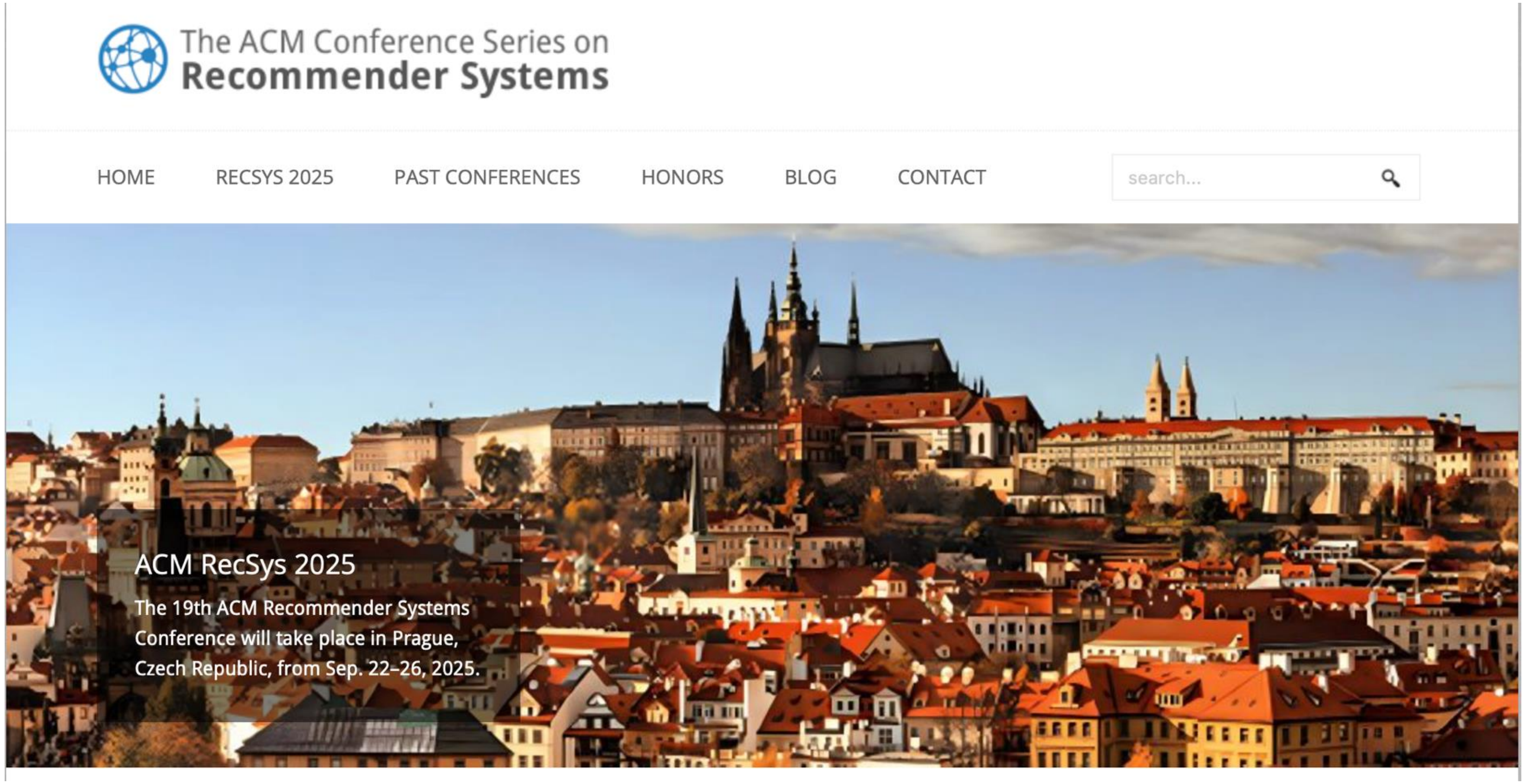


Before Recommendations **After Recommendations**

Figure 13. Each point is a user, and edge thickness is proportional to the pair's similarity

ACM Conference on RecSys

- https://recsys.acm.org



Challenges in RecSys

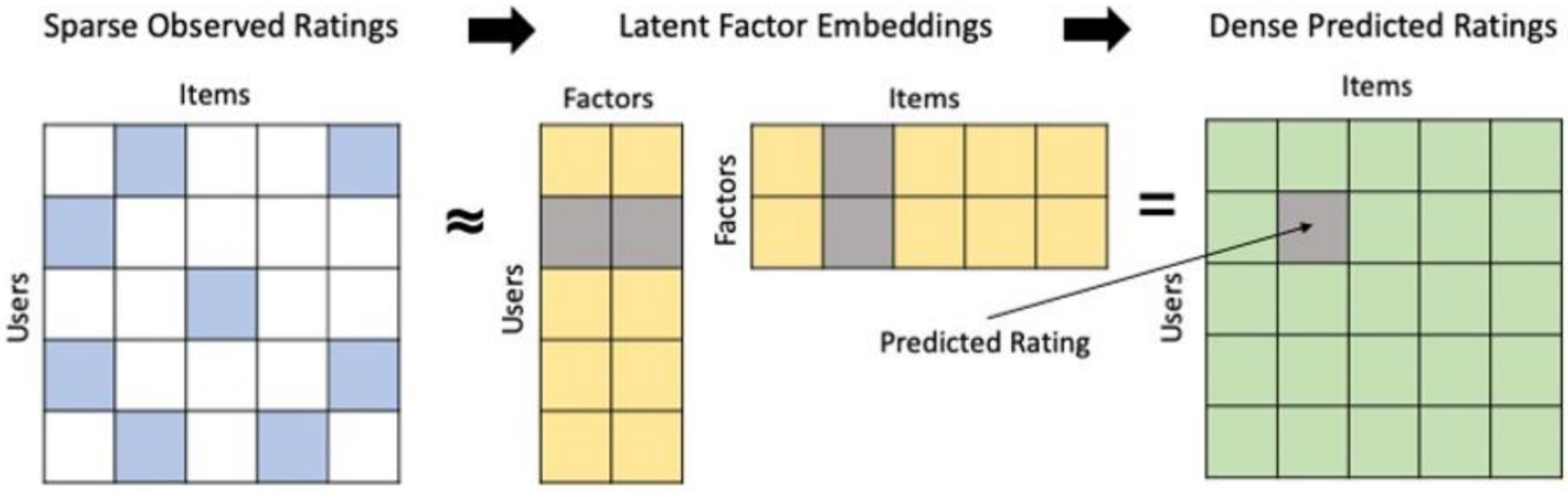
- Cold start problem
- Data sparsity
  - User count and item count are typically very large although the actual number of recommendations is very small
  - Users don’t rate all available items
- Scalability & latency Issues

Factorization Machine

- Motivation: challenges of traditional machine learning models in handling sparse data
- Proposed in Rendle, S. (2010). Factorization machines. 2010 IEEE International Conference on Data Mining
- A supervised algorithm
- Widely employed in modern advertisement and products recommendations

Factorized

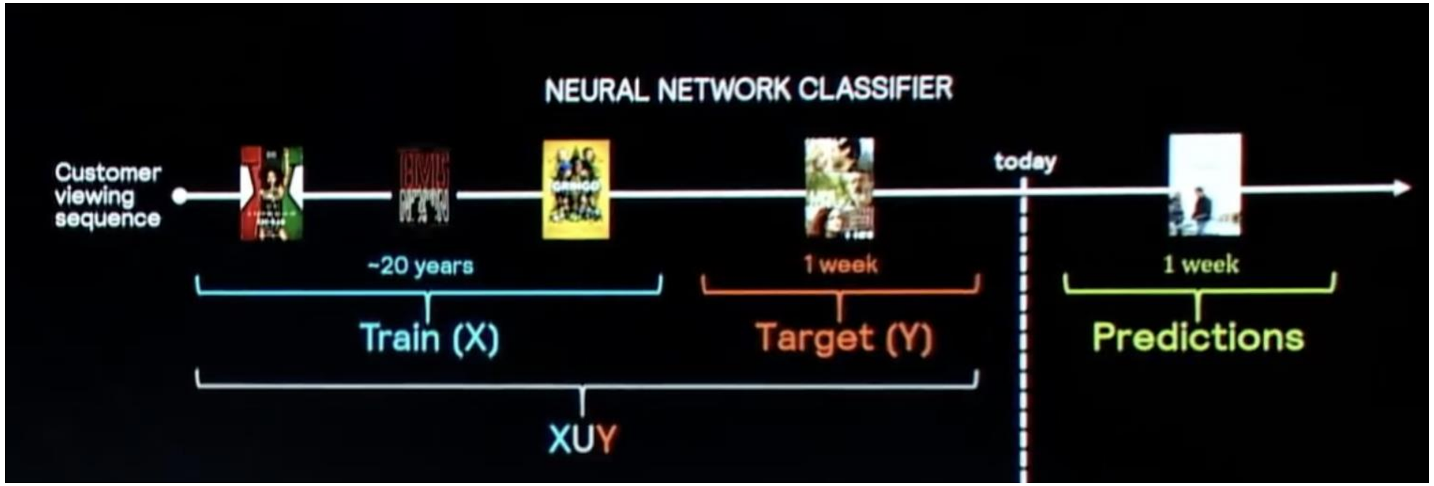
- Idea: reduce problem dimensionality using matrix factorization



https://medium.com/towards-data-science/factorization-machines-for-item-recommendation-with-implicit-feedback-data-5655a7c749db

Other Techniques

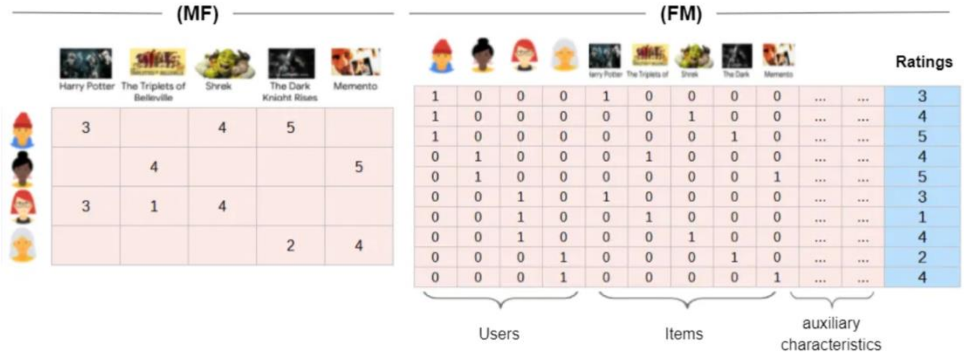
- Deep learning
  - https://youtu.be/GSQj27ps854?si=mWdSwyeDlcpOrEhX



- Reinforcement learning
  - Optimal recommendation policy by maximizing long-term user engagement and satisfaction

Lab 3

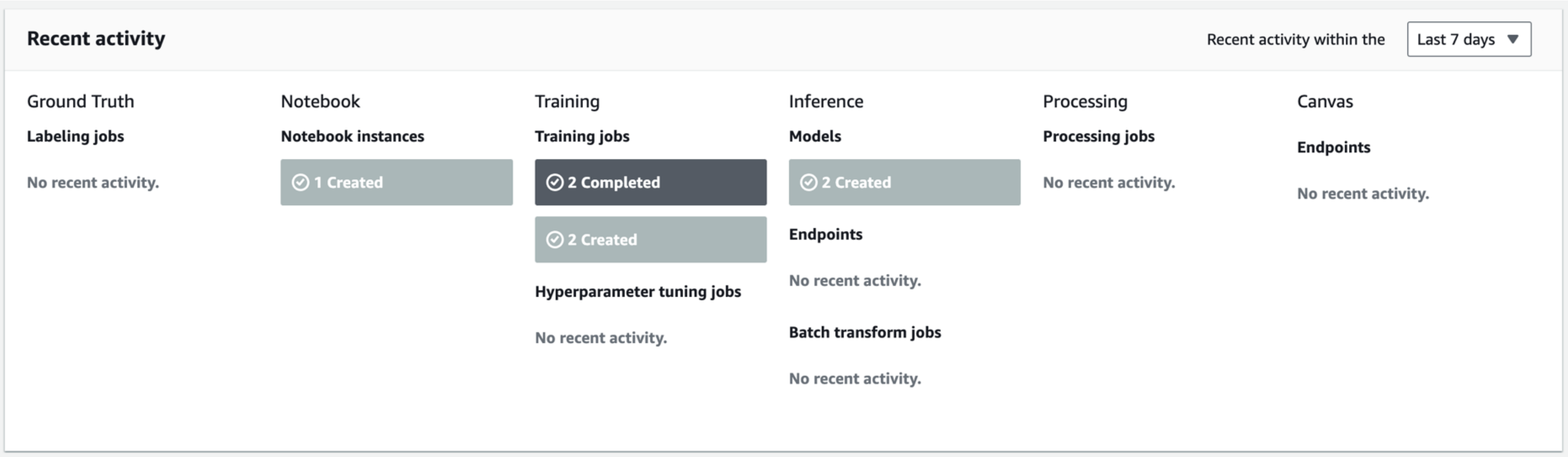
- Build a movie recommendation on MovieLens dataset
  - https://grouplens.org/datasets/movielens/
- We will use the classic ml-100K version
  - https://files.grouplens.org/datasets/movielens/ml-100k/
  - A tab separated list of **user id | item id | rating | timestamp**
  - Training set: **ua.base**; testing set: **ua.test;** 943 users on 1682 movies
- Use Factorization Machines built in AWS SageMaker



https://edransit.medium.com/factorization-machine-on-aws-the-best-algorithm-for-recommender-systems-53a250098030

Lab 3 (Cont.)

- Model deployment and hosting is very expensive
- **Make sure you delete all endpoints in the dashboard!!**
- Lab 3 cost: ~$1 if you terminate the resources correctly
- Always check SageMaker Dashboard once you are done



Learner Account Limitations

- This service can assume the LabRole IAM role
- Supported instance types: ml.t3.medium, ml.t3.large, ml.t3.xlarge, ml.m5.large, ml.m5.xlarge, ml.c5.large, ml.c5.xlarge only
- Maximum Sagemaker Notebooks: 2
- Maximum Sagemaker Apps: 2

- **Tips to preserve your budget**:
  - Choose the SageMaker dashboard link to view recent activity including running jobs, models, or instances. Stop or delete anything that is running and that you no longer need.
  - When using SageMaker Canvas or SageMaker Studio, logout of the session when you are done working with it. Consider deleting SageMaker Canvas and SageMaker Studio apps that are no longer needed.

Next Week

- Mining of massive datasets
- Computer vision and business applications

References

- https://aws.amazon.com/blogs/machine-learning/build-a-movie-recommender-with-factorization-machines-on-amazon-sagemaker/