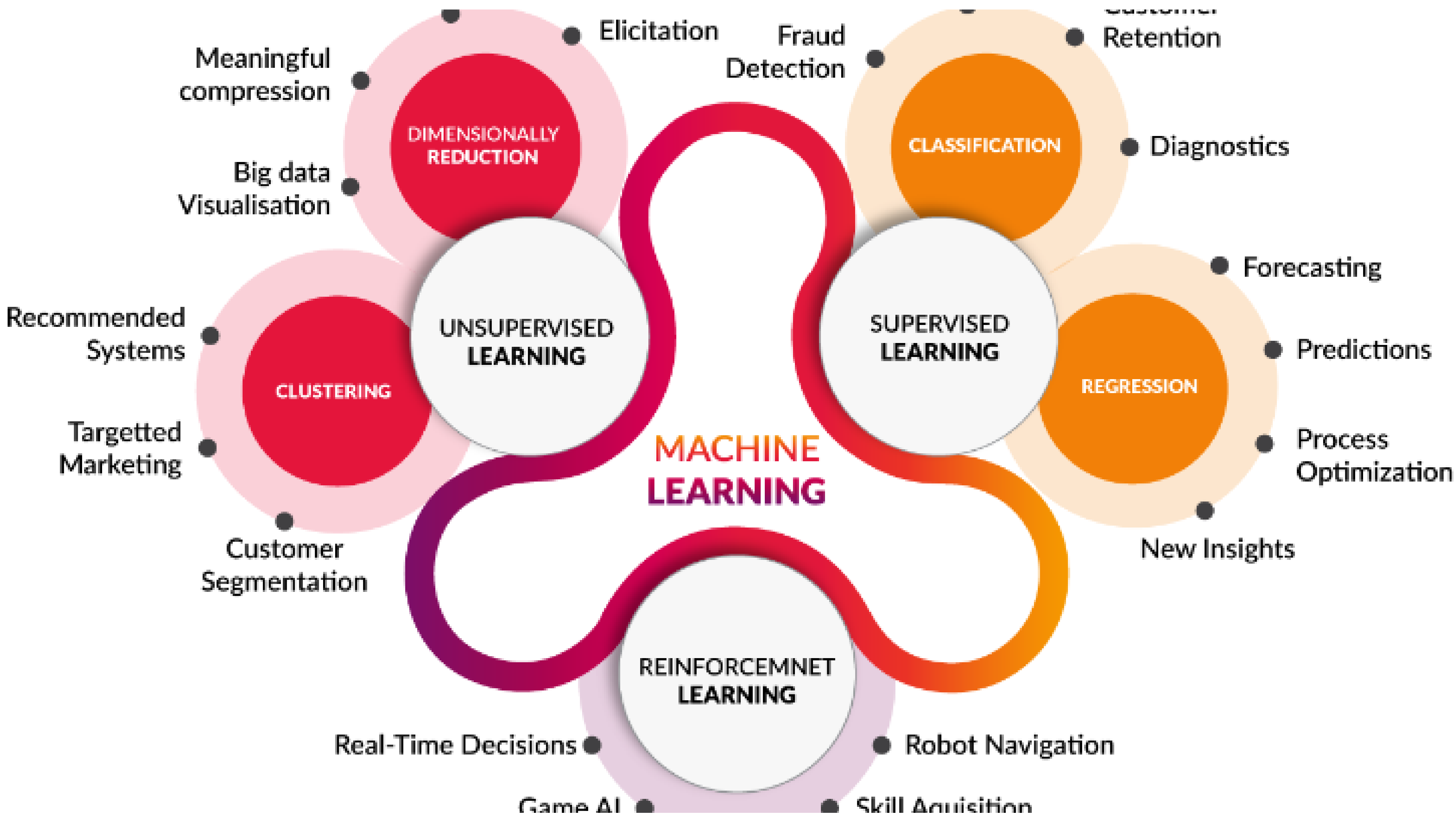
Recommendation Systems


Elicitation
Meaningful compression
DIMENSIONALLY REDUCTION
Big data Visualisation
Fraud Detection
Retention
CLASSIFICATION
Diagnostics
UNSUPERVISED LEARNING
SUPERVISED LEARNING
Forecasting
Recommended Systems
CLUSTERING
Predictions
REGRESSION
Targetted Marketing
Process Optimization
MACHINE LEARNING
Customer Segmentation
New Insights
REINFORCEMNET LEARNING
Real-Time Decisions
Robot Navigation
Game AI
Skill Aquisition

Why the coffee mug ?

A good recommendation is a one which does the job before the coffee in your cup is over.

- Someone must've said it

Recommender system

A **recommender system**, or a **recommendation system** (sometimes replacing 'system' with a synonym such as platform or engine), is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.[1][2] They are primarily used in commercial applications.

- Credits :

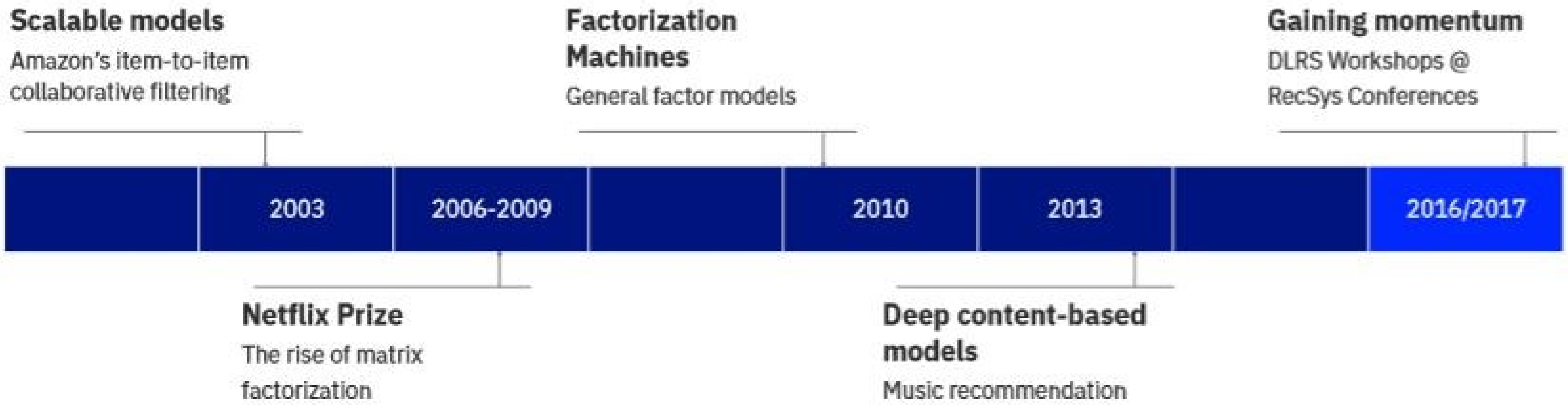
Recommender system

A **recommender system**, or a **recommendation system** (sometimes replacing 'system' with a synonym such as platform or engine), is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.[1][2] They are primarily used in commercial applications.

- Credits :

Evolution of Recommendation Models

- Netflix prize created significant progress (not unlike ImageNet)
- Matrix factorization techniques became SotA
- Restricted Boltzmann Machine (a class of NN) was one of the strongest single models
- Mostly tweaks until introduction of Factorization Machines
- Initial work on applying DL focused on feature extraction for content
- Huge momentum in DL techniques over the past two years



Types of Recommender Systems :

- Collaborative filtering
- Content-based filtering
- Multi-criteria recommender systems
- Risk-aware recommender systems
- Mobile recommender systems
- Hybrid recommender systems

Users and Items

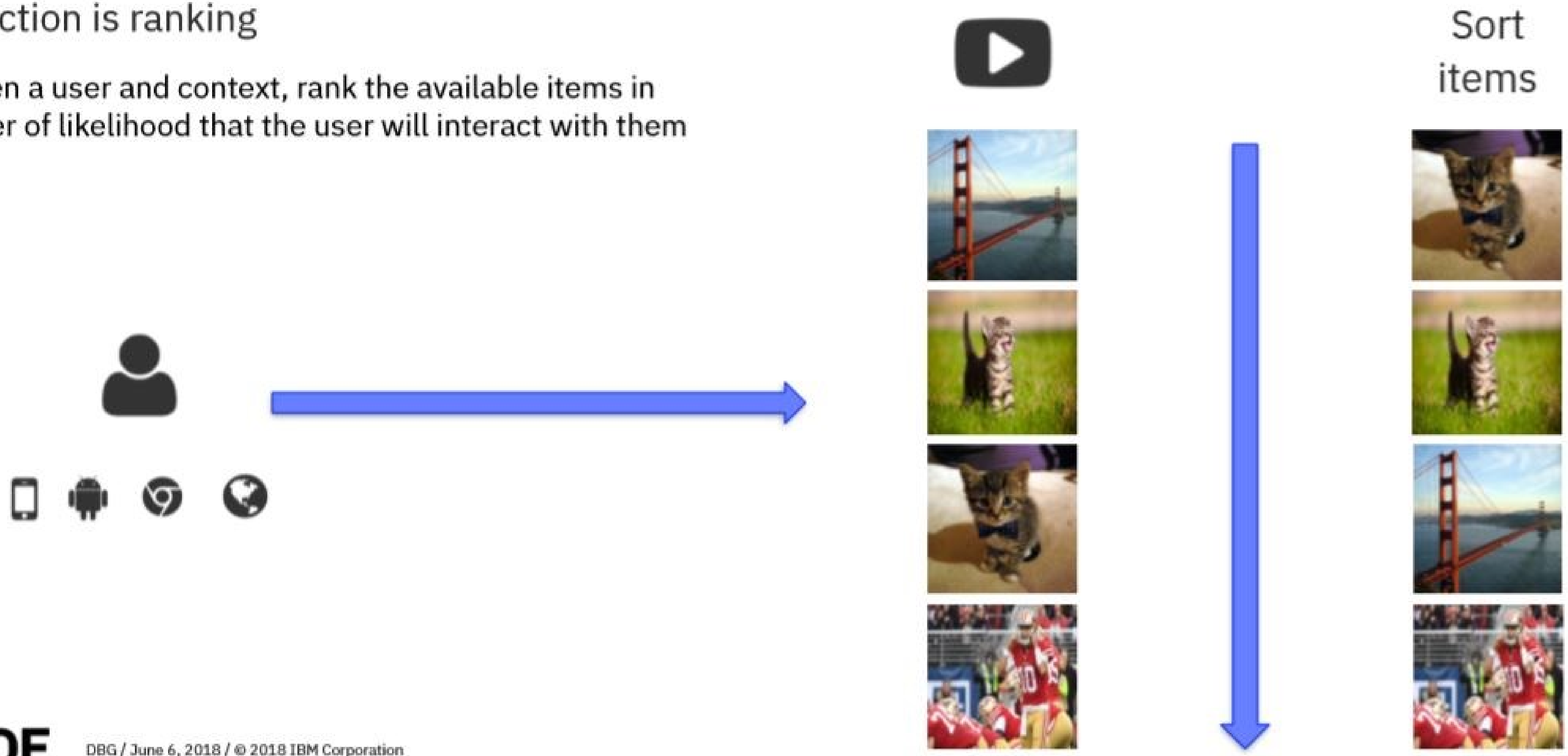
```
{
  "user_id": "1",
  "name": "Joe Bloggs",
  "created_date": 1476884080,
  "updated_date": 1476946916,
  "last_active_date": 1476946962,
  "age": 32,
  "country": "US",
  "city": "San Francisco",
  ...
}
```

```
{
  "item_id": "10",
  "name": "LOL Cats",
  "description": "catscatscats",
  "category": ["Cat Videos", "Humour", "Animals"],
  "tags": ["cat", "lol", "funny", "cats", "felines"],
  "created_date": 1476884080,
  "updated_date": 1476884080,
  "last_played_date": 1476946962,
  "likes": 100000,
  "author_id": "321",
  "author_name": "ilikecats",
  "channel_id": "CatVideoCentral",
  ...
}
```

Prediction

Prediction is ranking

- Given a user and context, rank the available items in order of likelihood that the user will interact with them



Categories :

Content based	User based	Popularity based
• Content drive. • User data involved – ratings from users or visiting a page by clicking a link. • Not good at capturing event-driven inter dependencies and/or complex behaviors. • Supervised Machine learning	• People like you also liked ___ . • Efficient when number of users is smaller than number of items up for recommendation. • Context independent • More accurate than content based .	• Most popular, best sellers, latest products, offers, location based popularity. • Not entirely data science but mostly based on surface level attributes. • Match visitors current browsing and what he looked at. • What is generally popular now.

Case-Based/Stereotype-Based

- Acquire info about user
- Classify user in a bucket (as a particular "case" or stereotype) based on facts about user
  - Eg soccer moms, poor grad student, … (there may be a hierarchy, rather than list of stereotypes)
- Certain assumptions about what appeals to a certain stereotype
  - Eg which items appeal to certain case/category of users
- Recommend those to the user

- Example: demographics-based recommendations

Feature-based/Content-based Filtering

- One approach: learning from item examples
  - Look at all items a user likes
    - Features of items
  - Find patterns among items and generalize (often also involves clustering)
  - Then recommend more items that fit same pattern(s)

  - Eg recommend movies based on features of those movies (genre, actors, …)

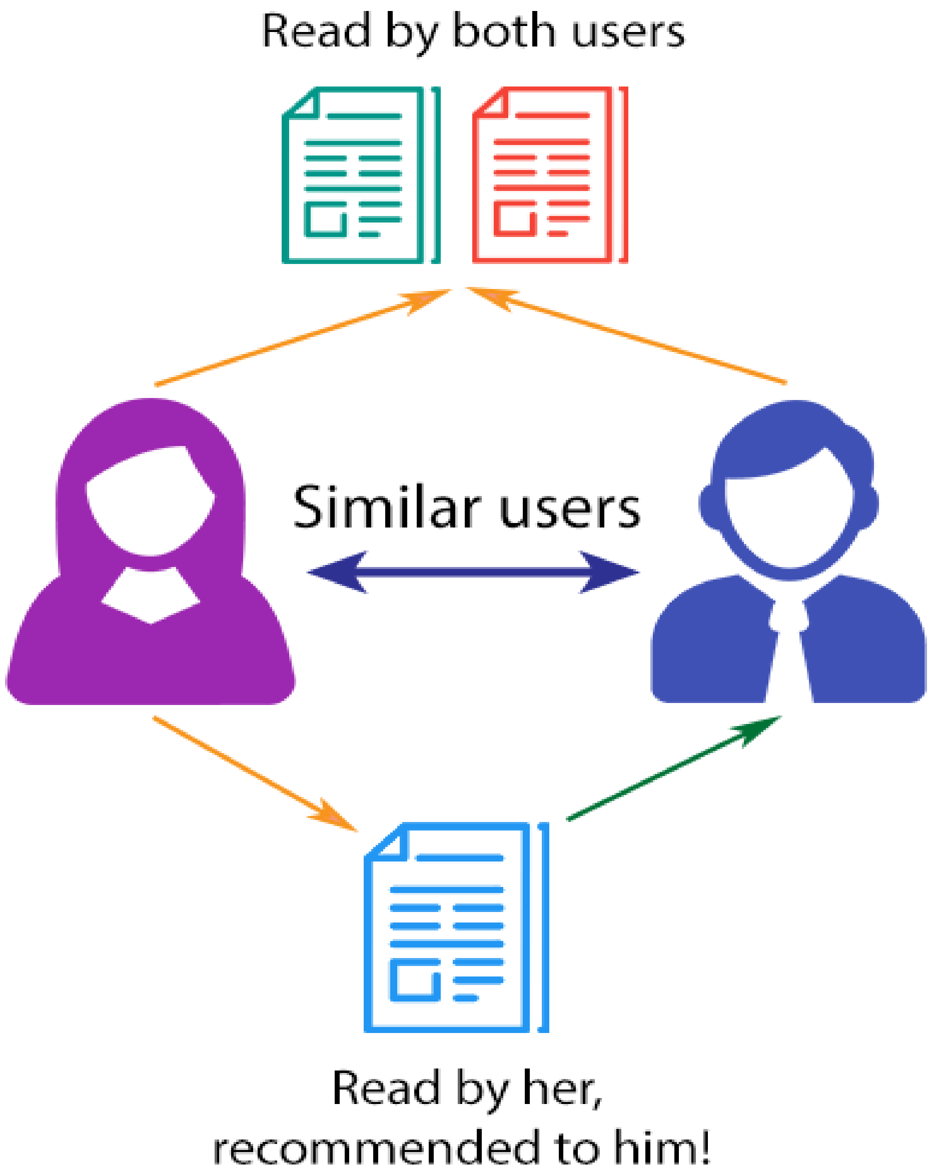
Feature-based/Content-based Filtering

- Another approach: learning stereotypes from user examples
  - Given a category of items
  - Given set of users with features & values
  - Given information on which users like what items
  - Generalize on what types of users like a category of items
  - Recommend items based on what case user falls into

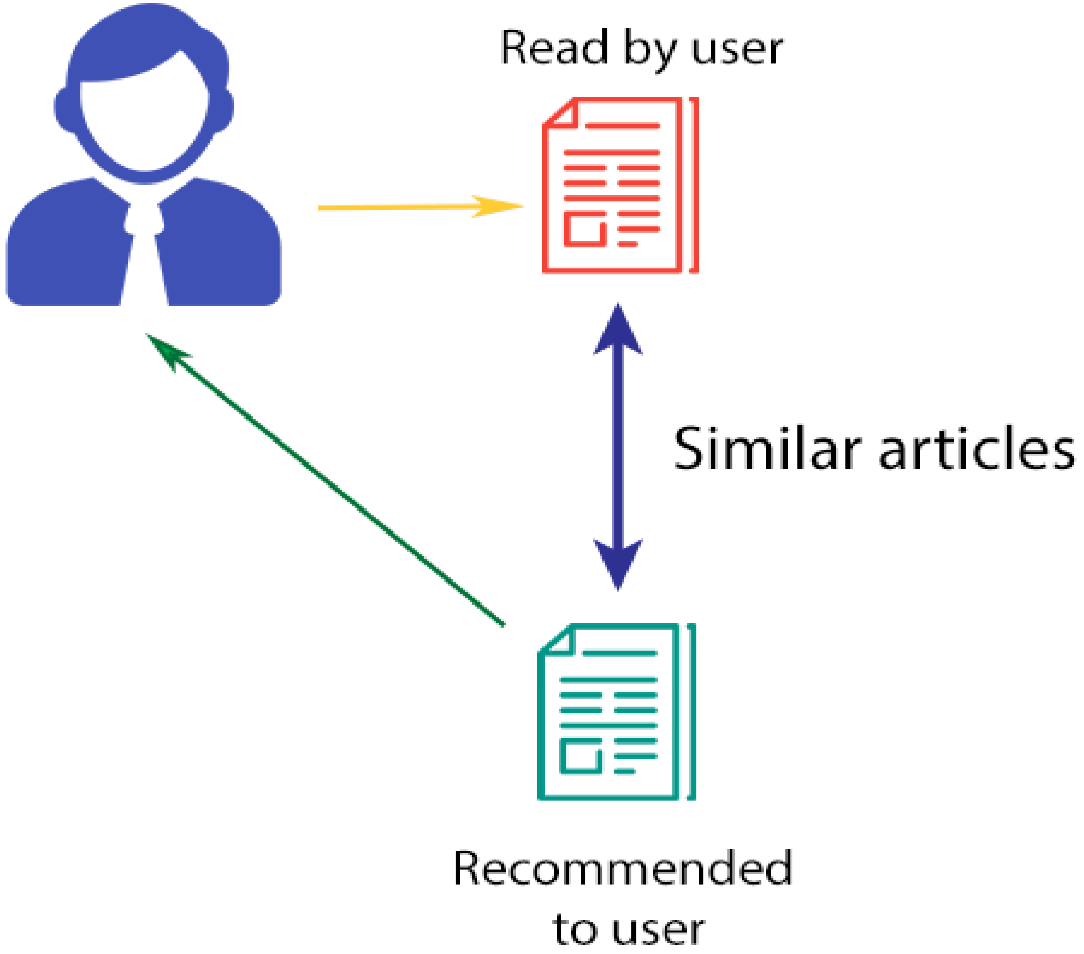
Knowledge-based Techniques

- Special case of feature-based where background knowledge of item space or user space is used to generalize
- Eg use ConceptNet or Interest Map
  - Know what people are related (InterestMap)
  - Know what items are related (ConceptNet)

COLLABORATIVE FILTERING



CONTENT-BASED FILTERING



Collaborative filtering

Collaborative Filtering

users

items

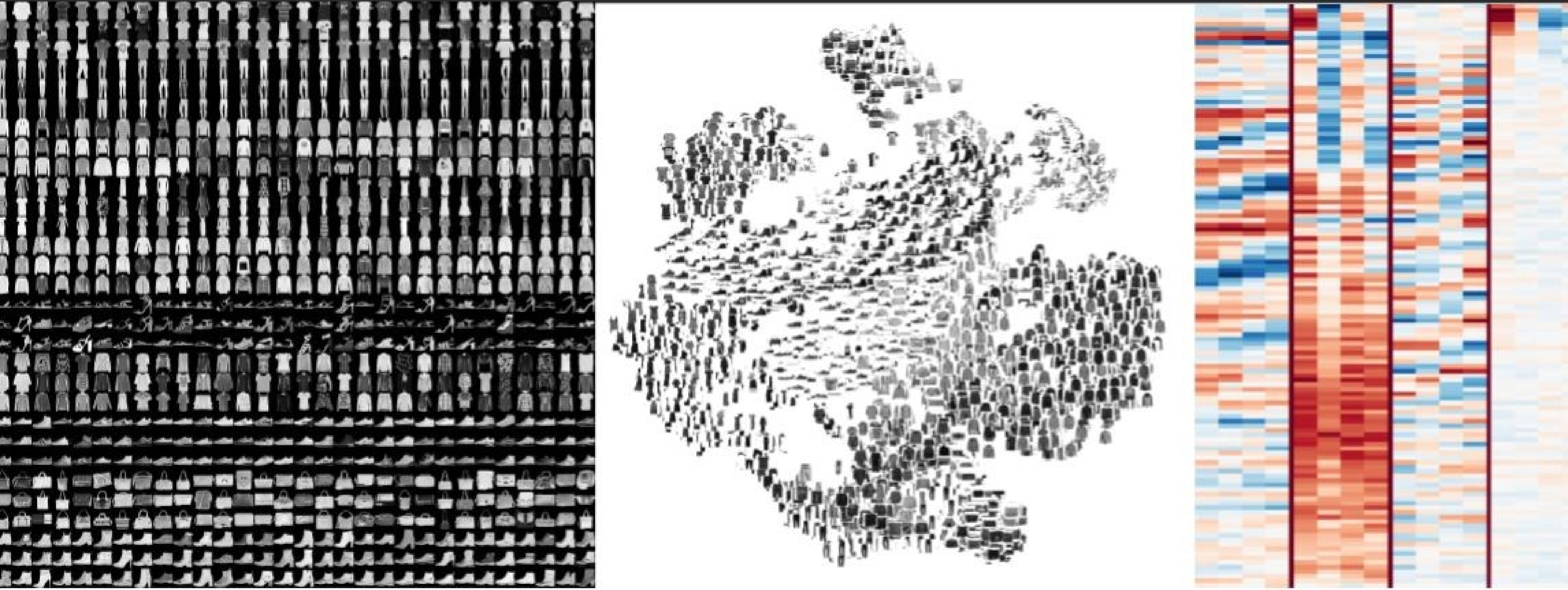
Sparse matrix of ratings (or purchase patterns)

Algorithms: recommend items based on item similarities (rows) or based on user similarities (colums), typically weighted average of K nearest neighbors, with weight inverse proportional to distance

Pros & Cons different techniques

- Collaborative filtering
  - Pros:
    - Does not require analysis of the items (features)
    - Better at qualitative judgements
  - Cons:
    - Bootstrapping
    - Ratings required
    - Critical mass required

Deep Learning for Recommendations

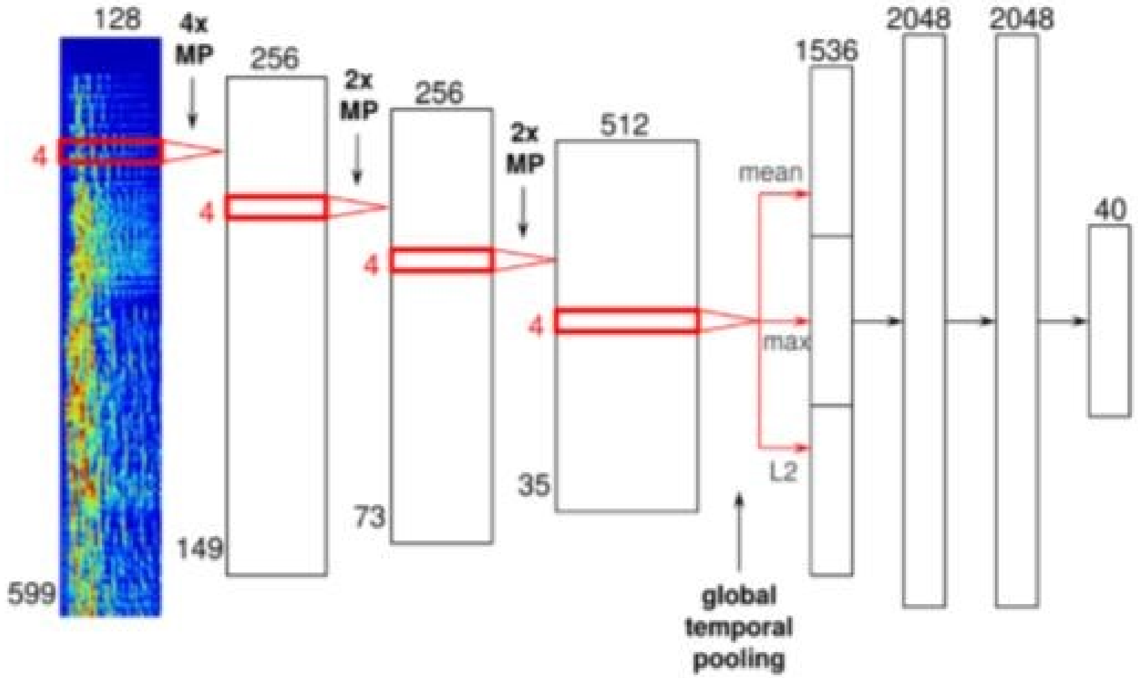


Source: Spotify / Sander Dieleman

Deep content-based music recommendation

Example of using a neural network model to act as feature extractor for item content / metadata

- CNN with audio spectogram as input data
- Filters capture lower-level audio characteristics, progressing to high-level features (akin to image problems)
- Max pooling and global pooling
- Fully connected layers with ReLU activations
- Output layer is the factor vector for the track from a trained collaborative filtering model
- Models trained separately

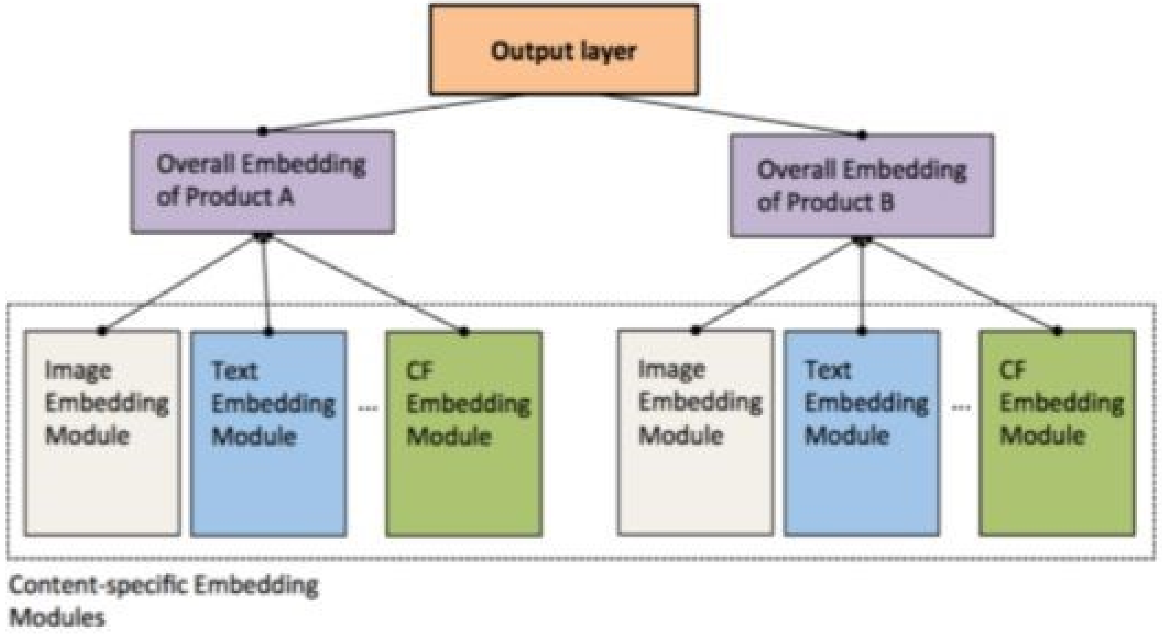


Source: Criteo Research

Content2Vec

Specialize content embeddings for recommendation

- Combine modular sets of feature extractors into one item embedding
- e.g. CNN-based model for images (AlexNet)
- e.g. Word2Vec, sentence CNN, RNN for text
- e.g. Prod2Vec for embedding collaborative filtering (co-occurrences)
- Modules mostly pre-trained in some form
- Final training step then similar to “transfer learning”
- Use pair-wise item similarity metric (loss)



WHY COMPANIES IMPLEMENT

RECOMMENDER SYSTEMS?

IMPROVE RETENTION

Continuously catering to users' preferences makes them more likely to remain loyal subscribers of the service

INCREASE SALES

Various research show an increase in upselling revenue ranging from 10-50% caused by accurate "You Might Also Like" product recommendations

FORM HABITS

Serving accurate content can trigger cues, building strong habits and influencing usage patterns in customers

ACCELERATE WORK

Analysts can save up to 80% time when served tailored suggestions for materials necessary for their further research

WHAT THEY TRACK	HOW THEY USE IT
AMAZON	
Customers' past purchases Items customers have rated and liked Customers' purchases compared to similar purchases by other customers Items in customers' virtual shopping carts	To make future product recommendations
NETFLIX	
Users' ratings on movies and television shows	To produce additional movie, TV show and genre recommendations

Question?

Why do companies consider using a recommendation system?

Motivation – Why Recommender Systems?

- Recommendation systems are everywhere. Some examples of impact:
  - "Netflix values recommendations at half a billion dollars to the company" [netflix recsys]
  - "LinkedIn job matching algorithms to improves performance by 50%" [San Jose Mercury News]
  - "Instagram switches to using algorithmic feed" [Instagram blog]

Some popular examples

+ **35 %**

Revenue due to RS

amazon

+ **33.3 %**

Increase in monthly subscriptions thanks to RS

Spotify

+ **60 %**

Amount of clicks due to recommendations

YouTube

+ **23.7 %**

Increase in revenue after adopting RS

BEST BUY

Credits : Google image search, IBM seminar, code review, amitkaps

Challenges

Challenges particular to recommendation models

– Data size and dimensionality (input & output)

– Extreme sparsity

  • Embeddings & compressed representations

– Wide variety of specialized settings

– Combining session, content, context and preference data

– Model serving is difficult – ranking, large number of items, computationally expensive

– Metrics – model accuracy and its relation to real-world outcomes and behaviors

– Need for standard, open, large-scale, datasets that have time / session data and are content- and context-rich

– Evaluation – watch you baselines!

  • [When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation]