



**Department of Algorithms and
Programming Technologies**



Recommender Systems



Alexey Dral

BigDATAteam



Evgeny Frolov

APT dept., FIFT MIPT

Moscow, November-December 2017

Outline (Lectures)

1. Introduction into RecSys (RS). RS classification.
Non-Personalised RS. Content-Based RS.
2. Collaborative Filtering RS. UU and II CF. Explicit & Implicit Feedback. Matrix Factorisation for RS
3. Advanced RS (context-aware, tensor decomposition, ...)

Outline (Homeworks)

- [1 week] Building simple RS (Non-personalized, CB)
- [2 weeks] RS contest (expected 2 tracks: prediction / recommendation task)

Introduction into RS: Goals

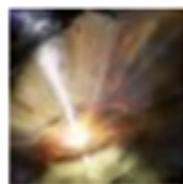
- **explain** basic ideas of Content-Based (**CB**), Collaborative Filtering (**CF**) and Knowledge-Based (**KB**) Recommender System algorithms;
- know steps to **train** simple RS algorithms;
- learn how to **design** RS experiment and **evaluate** it;
- learn how to **overcome** cold-start problem;

Introduction into RS: Outline

- **RS Basics: domains, classification**
- Non-personalised RS
- Content-Based RS
- RS Evaluation

RS: news

НОВОСТИ НАУКИ



[Ученые обнаружили «временной туннель» в центре галактики](#)

В центре галактики Млечный Путь находится не черная дыра, а так называемая «кротовья нора», с помощью которой якобы можно путешествовать сквозь пространство и время. Об этом заявил академик Российской академии наук (РАН) Николай Кардашев.

[Ученые определили фрукт, способный предотвратить рак](#)

[Ученые научились читать мысли человека при помощи смартфона](#)

[Стивена Хокинга выписали из больницы](#)

[Страдающие ожирением люди считают свой лишний вес нормой](#)

КОМПЬЮТЕРНЫЕ НОВОСТИ



[Появилось первое фото нового смартфона с клавиатурой от BlackBerry](#)

Непосредственно перед уходом производитель выпустит новый смартфон, оснащенный аппаратной клавиатурой, который получил название Mercury. В сети появились снимки нового смартфона, на которых можно рассмотреть лицевую панель BlackBerry Mercury.

[Безопасность](#)

[Соф트](#)

[Hardware](#)

[Интернет](#)

[Игры](#)

[Hisense A2 с E-Ink-экраном на задней панели замечен в TENAA](#)

[Беспроводной интернет празднует 25-летие](#)

[СМИ: Новый iPad от Apple избавится от кнопки «Домой»](#)

[Microsoft обновила функцию символьных ссылок в Windows 10](#)

RS: e-commerce

Хиты продаж



Электромясорубка VITEK
VT-3614 G

★★★★★ (3)

4990р.
2990р.



Электрочайник Scarlett SC
- EK18P28

★★★★★ (23)

1390р.
990р.



Электрочайник Maxwell
MW-1041 GD

★★★★★ (19)

1990р.
1290р.



RS: movie

Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes Celebs, Events & Photos News & Community Watchlist

recommended movies

by [ordanzala](#) created 25 Oct 2013 | last updated - 3 months ago

Page 1 of 8 (708 Titles) Sort by: List Order (ascend) View:

[Log in](#) to copy items to your own lists.

1. **Бёрдмэн** (2014)
★★★★★ ★★★★★ 7.8/10
Illustrated upon the progress of his latest Broadway play, a former popular actor's struggle to cope with his current life as a wasted actor is shown. (119 mins.)
Director: [Alejandro G. Iñárritu](#)
Stars: [Michael Keaton](#), [Zach Galifianakis](#), [Edward Norton](#), [Andrea Riseborough](#)
[Add to Watchlist](#)

2. **Побег из Шоушенка** (1994)
★★★★★ ★★★★★ 9.3/10
Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency. (142 mins.)
Director: [Frank Darabont](#)
Stars: [Tim Robbins](#), [Morgan Freeman](#), [Bob Gunton](#), [William Sadler](#)

RS: music

Слушают сейчас



	Lordly feat. Alex Aiono Feder	♡ + ⬆
	My Way Calvin Harris	3:39
	Alone Alan Walker	2:41
	Lost on You Swanky Tunes & Going Deeper Remix; Radio Edit L.P.	3:30
	Туманы Макс Барских	3:26
	Твои глаза LOBODA	3:53
	We Don't Talk Anymore feat. Selena Gomez Charlie Puth	3:37

RS: what can it be?



RS: domains

- news
- e-commerce
- movie
- music
- finance
- life insurance
- tourism
- job search
- real estate
- ...

RS: domains

- news
- e-commerce
- movie
- music
- finance
- life insurance
- tourism
- job search
- real estate
- ...

extras: cross-domain

RS: categories

- Non-Personalised
- Content-Based (CB)
- Neighbourhood Models (UU / II CF)
- Collaborative Filtering (CF)
- Knowledge-Based (KB)
- + Hybrid Recommender Systems

Non-Personalised

most popular items

Top Tech Products & Gadgets

1–6 of 6



Samsung Galaxy S8



Nintendo Switch



Sony Alpha a6000



Fitbit Charge 2



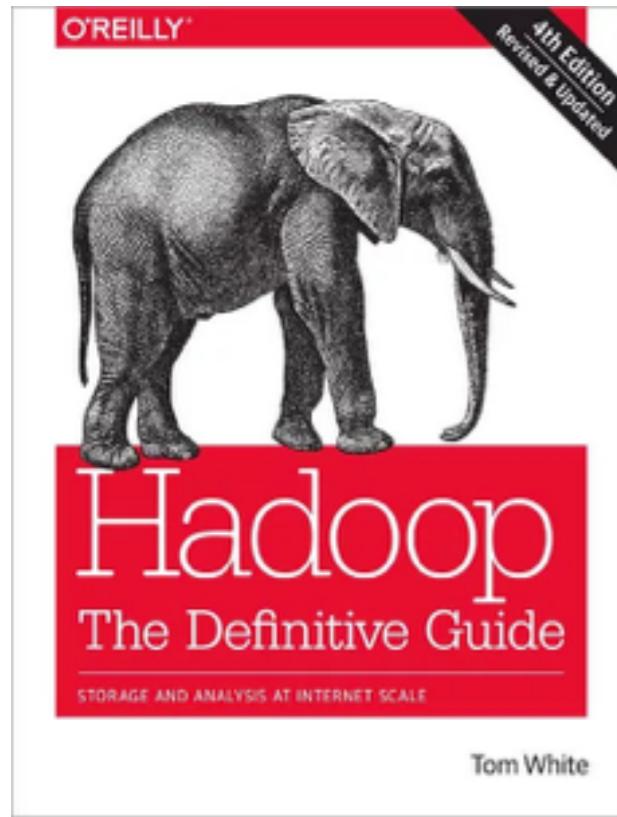
Apple MacBook Pro
with Touch Bar



Fidget Spinners

from: <https://www.google.com/shopping?hl=en>

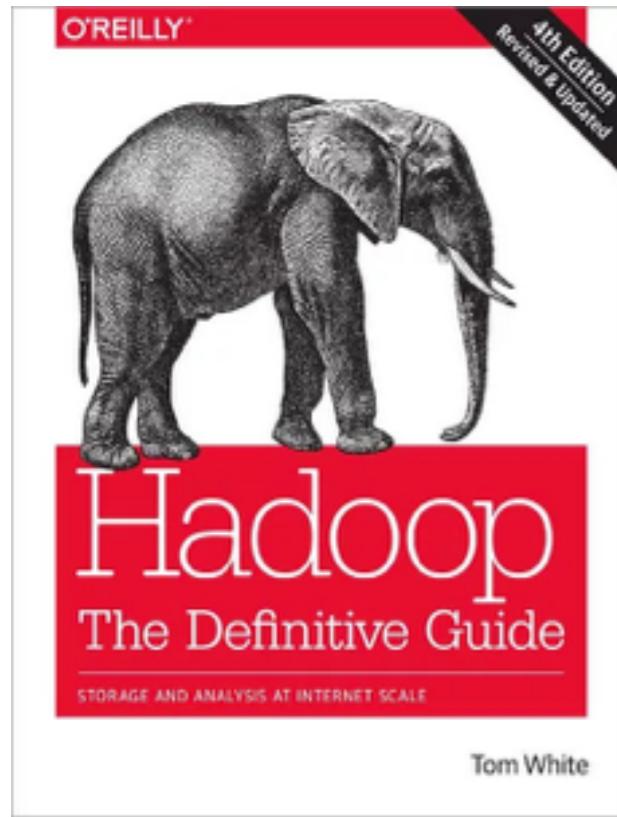
Content Based (CB)



Property	Expected Type	Description
Properties from Book		
bookEdition	Text	The edition of the book.
bookFormat	BookFormatType	The format of the book.
illustrator	Person	The illustrator of the book.
isbn	Text	The ISBN of the book.
numberOfPages	Integer	The number of pages in the book.

user	book	author	...	target
programmer A	C++	Bjarne Stroustrup	...	likes
programmer A	Cooking	Gordon Ramsay	...	dislikes
programmer B	Cooking	Gordon Ramsay	...	likes
...

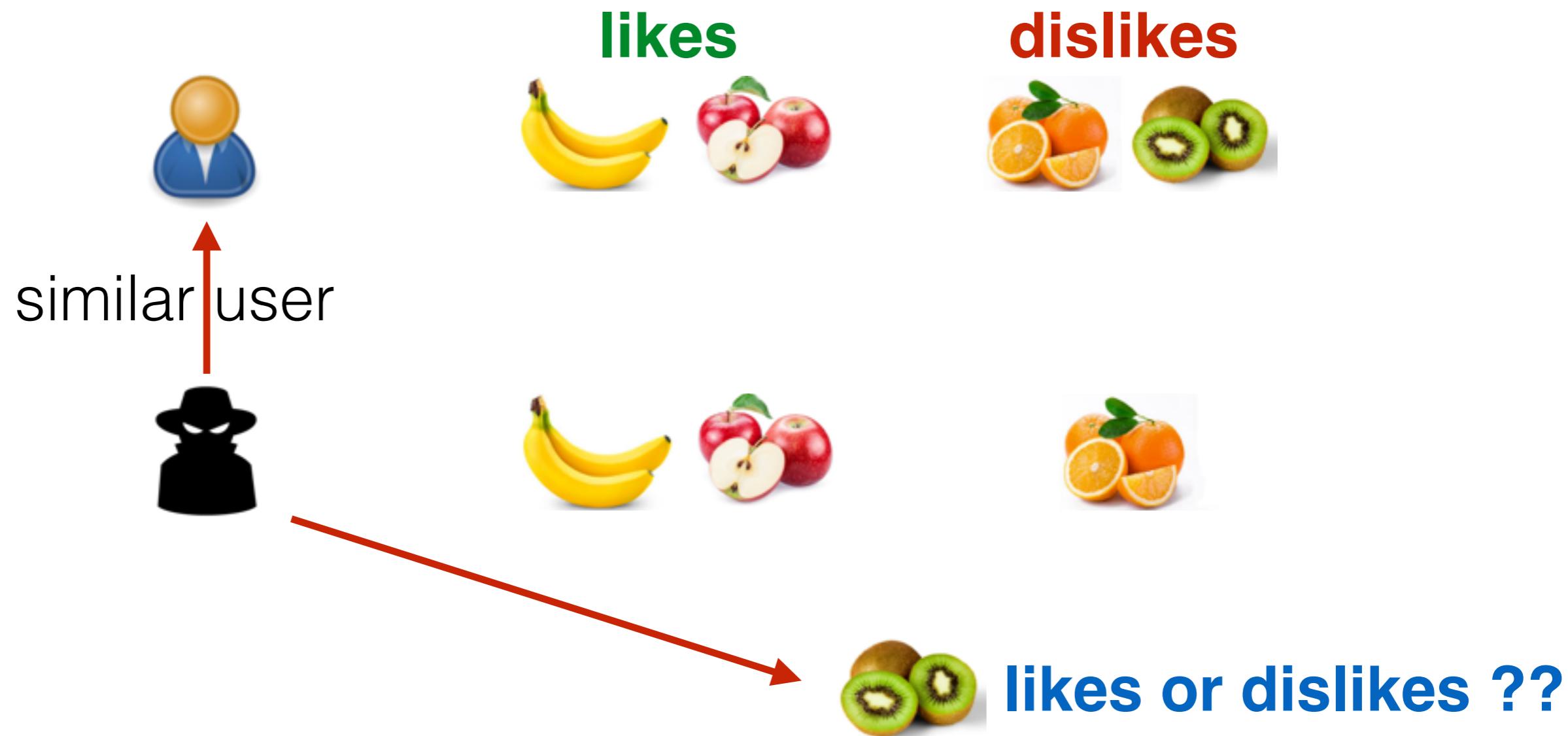
Content Based (CB)



Property	Expected Type	Description
Properties from Book		
bookEdition	Text	The edition of the book.
bookFormat	BookFormatType	The format of the book.
illustrator	Person	The illustrator of the book.
isbn	Text	The ISBN of the book.
numberOfPages	Integer	The number of pages in the book.

user	book	author	...	target
programmer A	C++	Bjarne Stroustrup	...	likes
programmer A	Cooking	Gordon Ramsay	...	dislikes
programmer B	Cooking	Gordon Ramsay	...	likes
...
programmer A	Linux Cookbook	Carla Schroder	...	???

Neighbourhood Models (User-User / Item-Item)



Neighbourhood Models (User-User / Item-Item)

purchase history (yesterday)



Recommender
System



Do you want to buy ??



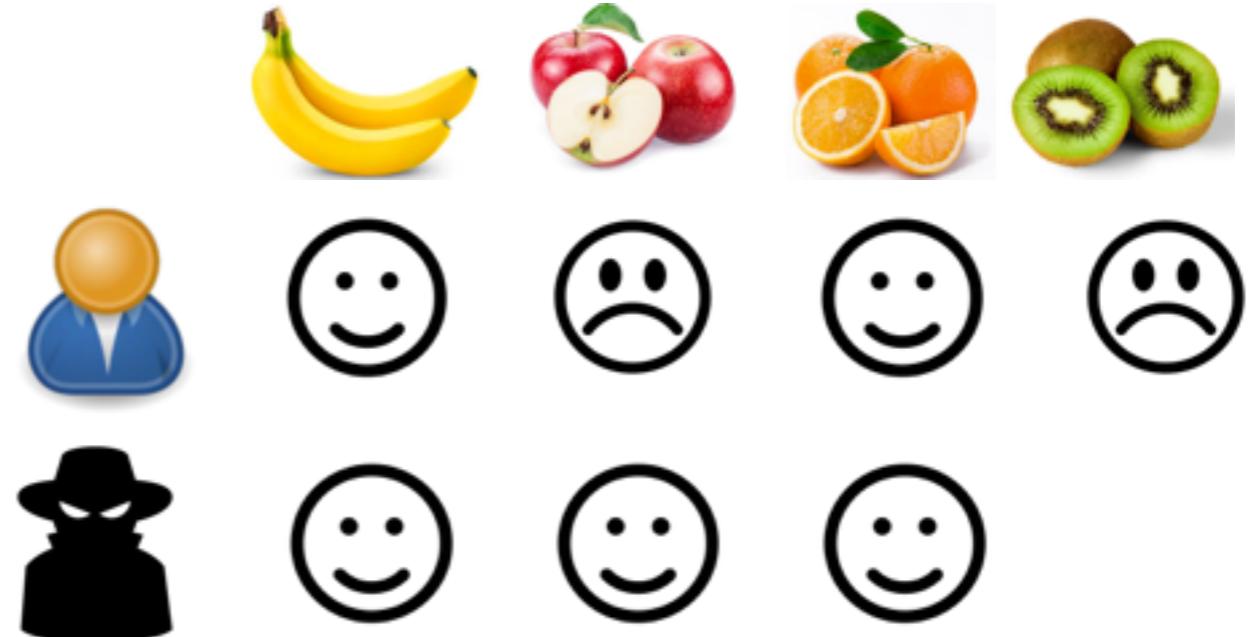
Collaborative Filtering (CF)



2006, October, 2nd (start)
2007 - progress Prize
2008 - progress Prize
2009 - June/July 25, 10+%
(improvement)

\$1M Grand Prize

Collaborative Filtering (CF)

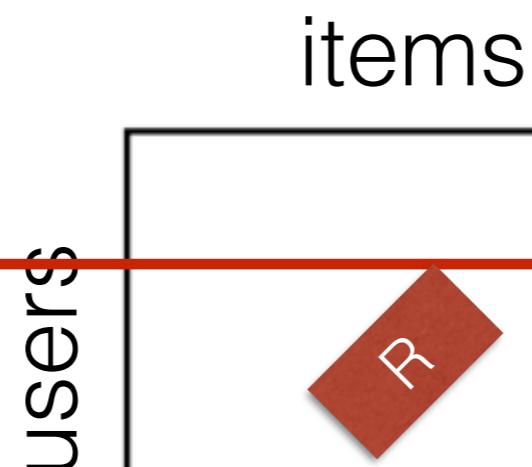


rating matrix ($R = PQ$)

\$1M Grand Prize

latent item matrix

latent user matrix



Knowledge Based (KB)



£165,000 - 2 bedroom ...
[First Avenue, Stafford ST16](#)



£1,325,000 - 4 bedroom terraced ...
[Howgate Road, London SW14](#)



£149,950 - 3 bedroom terraced ...
[Kendal Road, St Annes, ...](#)

Knowledge Based (KB)



£165,000 - 2 bedroom ...
[First Avenue, Stafford ST16](#)



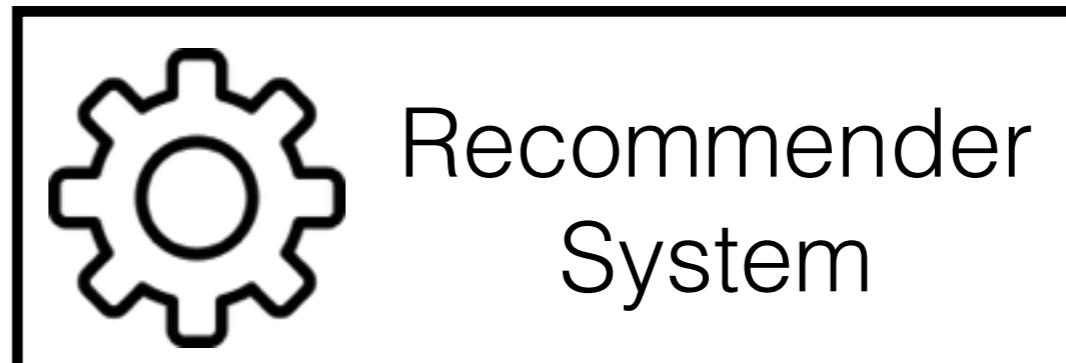
£1,325,000 - 4 bedroom terraced ...
[Howgate Road, London SW14](#)



£149,950 - 3 bedroom terraced ...
[Kendal Road, St Annes, ...](#)



get
suggestions



Hybrid RecSys: $\text{Alg}_1 + \text{Alg}_2 + \dots$

- news (**CB, CF**)
- e-commerce (**CF**)
- movie (**CF**)
- music (**CB / CF**)
- finance (**KB**)
- life insurance (**KB**)
- tourism (**CB, KB**)
- job search (**CB**)
- real estate (**KB**)
- ...

What types of Recommender Systems exist?

- Content-Based
- Collaborative-Based
- Content-Filtering
- Hybrid
- Knowledge-Based
- Collaborative Filtering

Evaluation

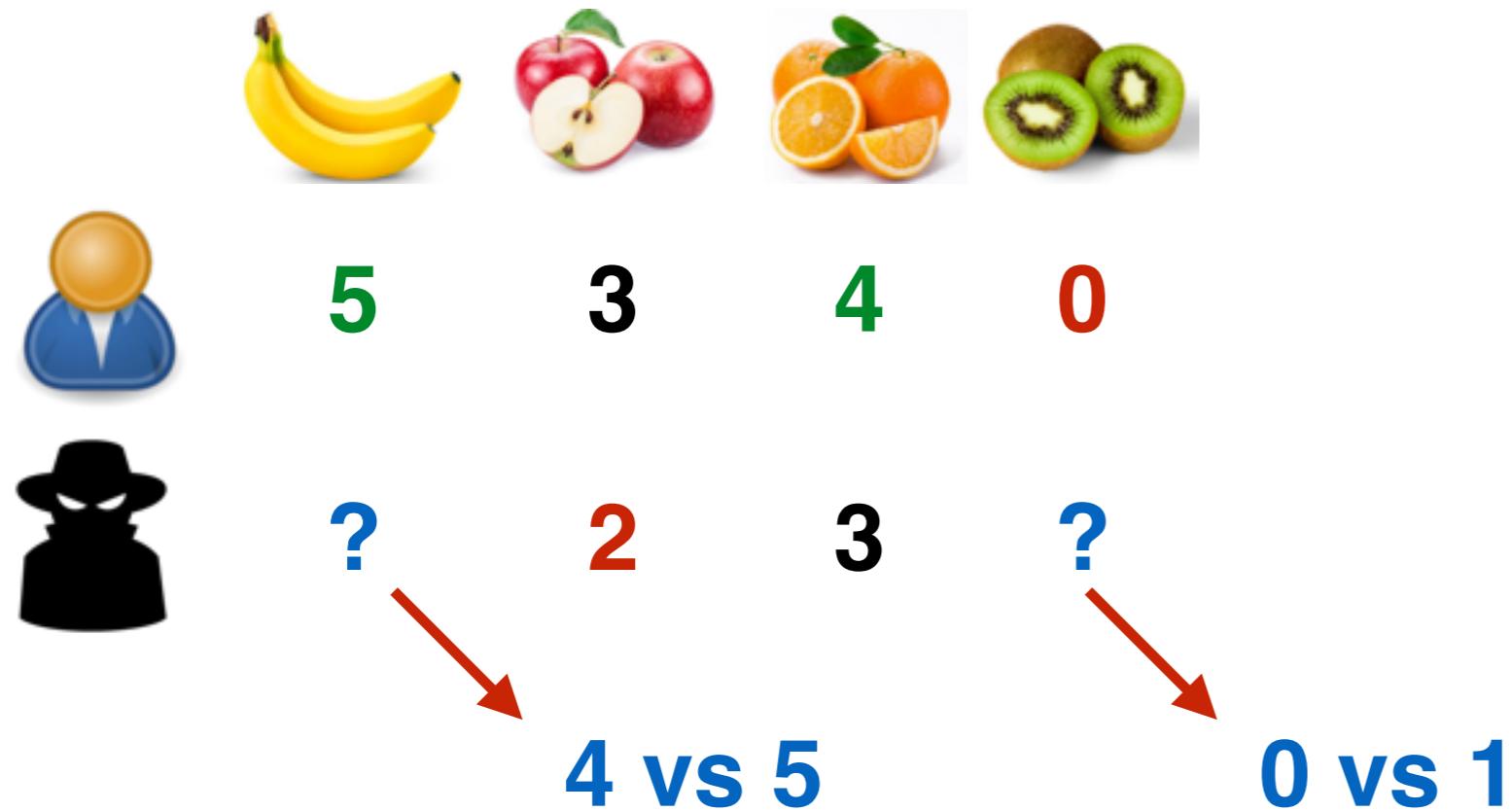


5 3 4 0

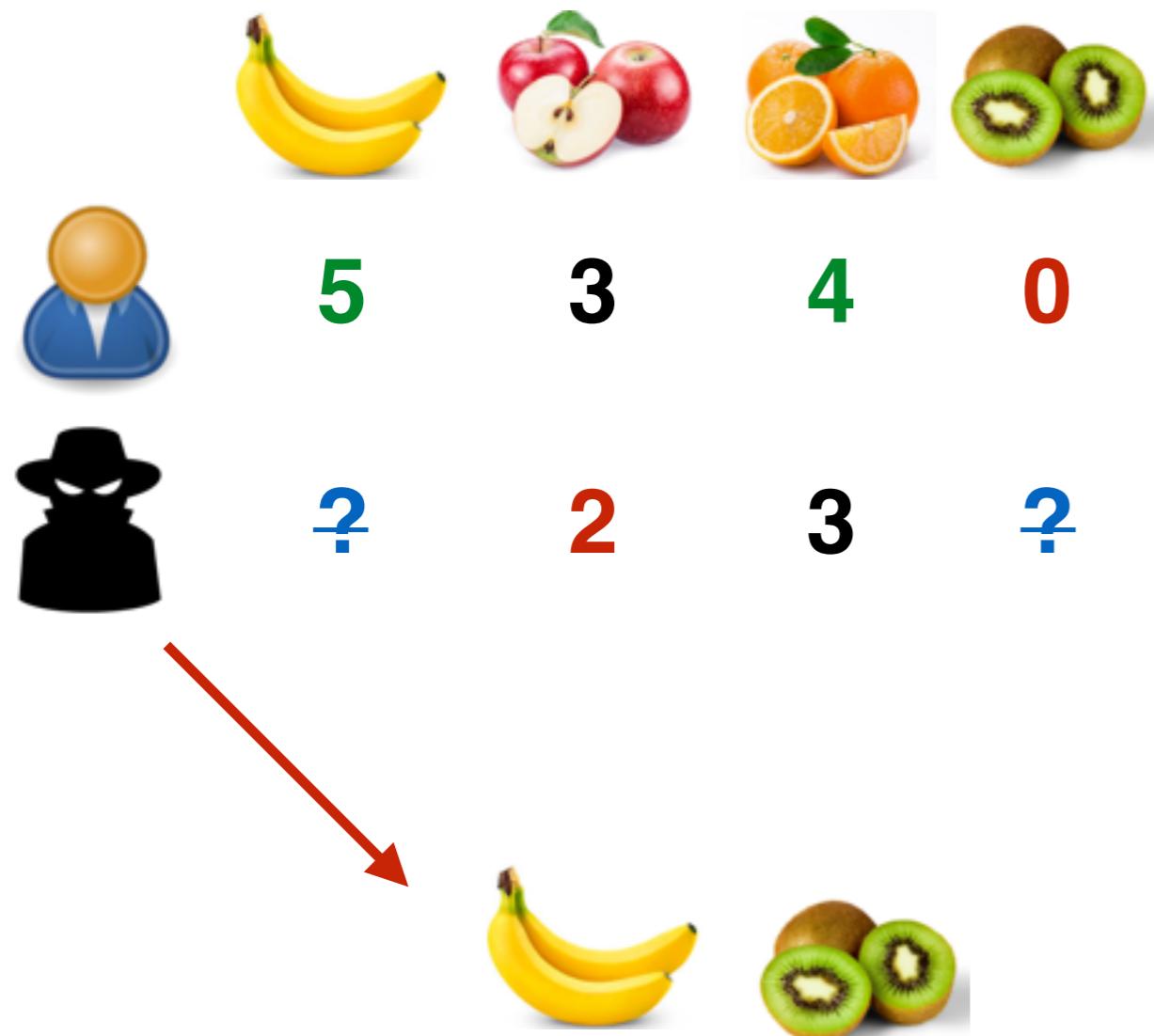


? 2 3 ?

Evaluation



Evaluation



order of recommendations

Prediction (metrics):

- MAE, RMSE, ...

Recommendation:

- MRR, DCG, ...

Evaluation



recommendations



season#5



season#3



season#9

...

Prediction (metrics):

- MAE, RMSE, ...

Recommendation:

- MRR, DCG, ...

Evaluation



recommendations



season#5



season#3



season#9

...

Prediction (metrics):

- MAE, RMSE, ...

Recommendation:

- MRR, DCG, ...

extras: coverage, novelty, diversity, serendipity, ...

RecSys Challenges

RecSys Challenges

- Cold-Start Problem

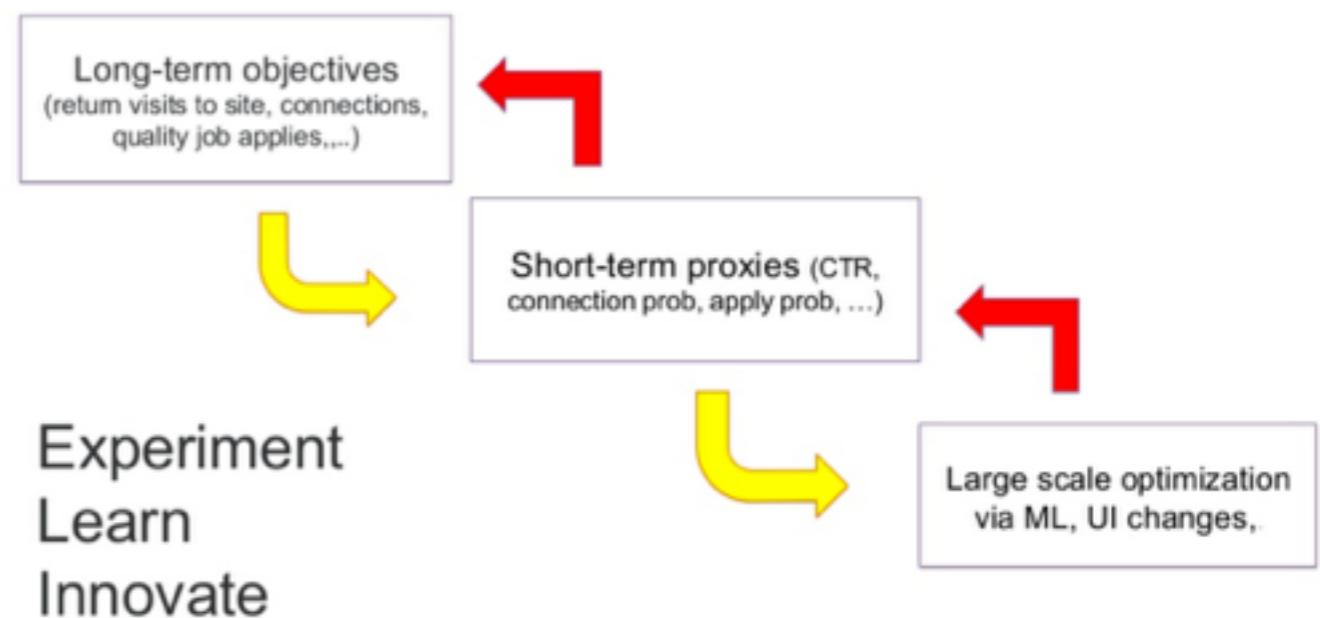


RecSys Challenges

- Cold-Start Problem
- Offline / Online correlation

RecSys'16 tutorial: Lessons learned from building real-life recommender systems

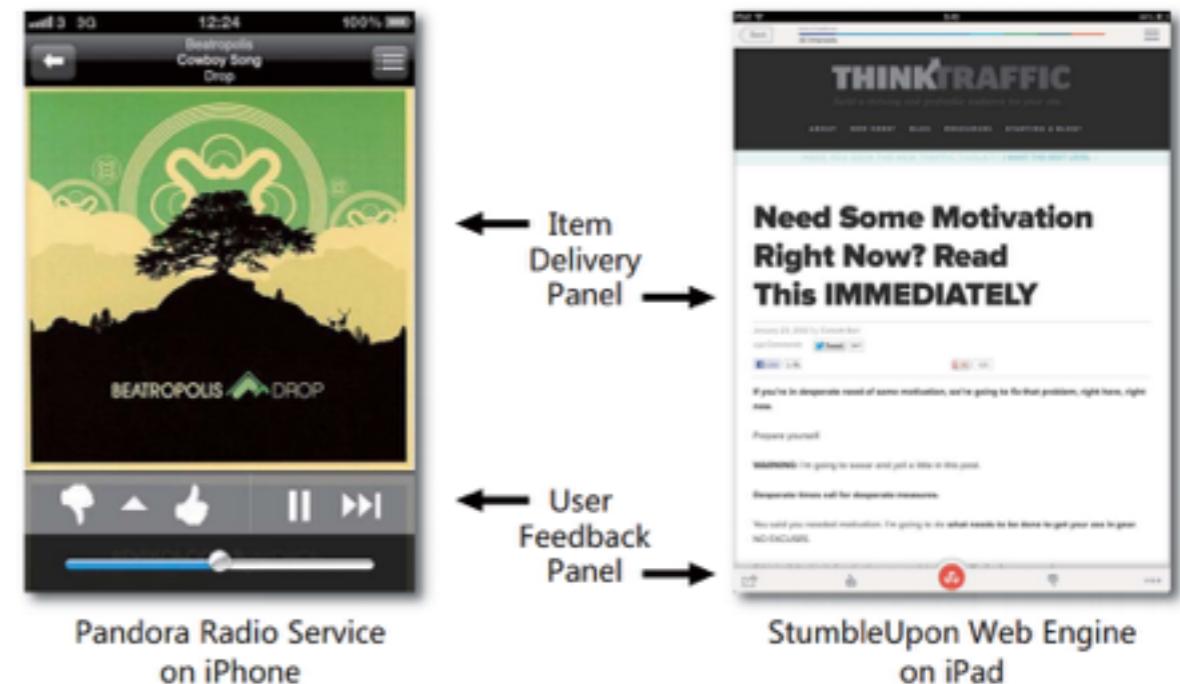
Connecting long-term objectives to proxies that can be optimized by machines/algorithms Clip slide



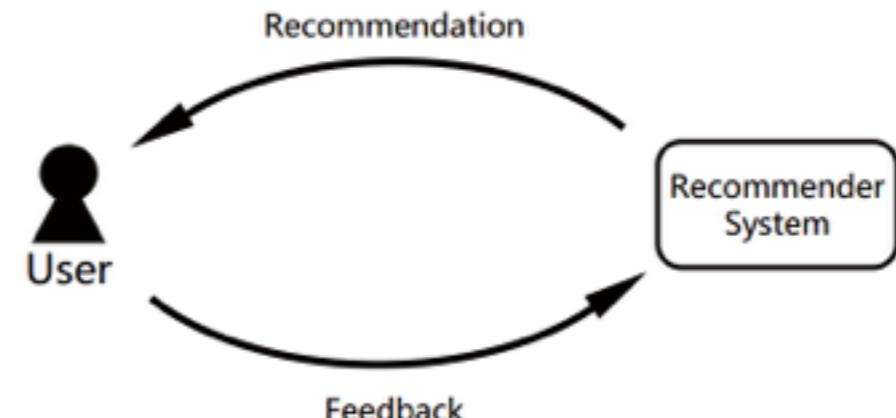
credit: Deepak Agarwal,
Senior Director of Engineering at LinkedIn

RecSys Challenges

- Cold-Start Problem
- Offline / Online correlation
- Interface Design, Exploration
Exploitation dilemma



(a) Two examples of Interactive Recommender Systems.



(b) Non-stop Recommendation-feedback loop.

RecSys Challenges

- Cold-Start Problem
- Offline / Online correlation
- Interface Design, Exploration / Exploitation dilemma
- Scalability
- Imbalanced Dataset (activity - power law distribution)

Outline (Introduction into RS)

- RS Basics: domains, classification
- **Non-personalised RS**
- Content-Based RS
- RS Evaluation



provided by



is a research lab in the
Department of Computer Science
and Engineering at the University of
Minnesota



movielens

```
$ head ~/path/to/ratings.csv  
userId,movieId,rating,timestamp  
1,31,2.5,1260759144  
1,1029,3.0,1260759179  
1,1061,3.0,1260759182  
...
```

```
$ head ~/path/to/movies.csv  
movieId,title,genres  
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy  
2,Jumanji (1995),Adventure|Children|Fantasy  
3,Grumpier Old Men (1995),Comedy|Romance  
4,Waiting to Exhale (1995),Comedy|Drama|Romance
```

```
from pyspark.mLLib import recommendation
from pyspark import SparkContext

def to_rating(ml_latest_row):
    user, movie, rating, timestamp = ml_latest_row.split(',')
    return recommendation.Rating(int(user), int(movie), float(rating))

def skip_header(rdd):
    header = rdd.first()
    rdd = rdd.filter(lambda row: row != header)
    return rdd

if __name__ == '__main__':
    sc = SparkContext(appName='RecSys Non-Personalized')
    data = sc.textFile("./path/to/ratings.csv")
    data = skip_header(data)
    movielens = data.map(to_rating)
    movie_ratings = movielens.map(lambda r: (r.product, r.rating)).groupByKey()
    movie_avg_rating = movie_ratings.map(lambda (movie_id, ratings): (movie_id,
        sum(ratings) / len(ratings)))
    movie_avg_rating.map(lambda (k, v): (v, k)).sortByKey(ascending=False).take(10)
```

```
from pyspark.mllib import recommendation
from pyspark import SparkContext

def to_rating(ml_latest_row):
    user, movie, rating, timestamp = ml_latest_row.split(",")
    return recommendation.Rating(int(user), int(movie), float(rating))

def skip_header(rdd):
    header = rdd.first()
    rdd = rdd.filter(lambda row: row != header)
    return rdd

if __name__ == "__main__":
    sc = SparkContext(appName="RecSys Non-Personalized")
    data = sc.textFile("./path/to/ratings.csv")
    data = skip_header(data)
    movielens = data.map(to_rating)
    movie_ratings = movielens.map(lambda r: (r.product, r.rating)).groupByKey()
    movie_avg_rating = movie_ratings.map(lambda (movie_id, ratings): (movie_id,
        sum(ratings) / len(ratings)))
    movie_avg_rating.map(lambda (k, v): (v, k)).sortByKey(ascending=False).take(10)
```

```
from pyspark.mllib import recommendation
from pyspark import SparkContext

def to_rating(ml_latest_row):
    user, movie, rating, timestamp = ml_latest_row.split(',')
    return recommendation.Rating(int(user), int(movie), float(rating))

def skip_header(rdd):
    header = rdd.first()
    rdd = rdd.filter(lambda row: row != header)
    return rdd

if __name__ == '__main__':
    sc = SparkContext(appName='RecSys Non-Personalized')
    data = sc.textFile("./path/to/ratings.csv")
    data = skip_header(data)
    movielens = data.map(to_rating)
    movie_ratings = movielens.map(lambda r: (r.product, r.rating)).groupByKey()
    movie_avg_rating = movie_ratings.map(lambda (movie_id, ratings): (movie_id,
        sum(ratings) / len(ratings)))
    movie_avg_rating.map(lambda (k, v): (v, k)).sortByKey(ascending=False).take(10)
```

```
from pyspark.mllib import recommendation
from pyspark import SparkContext

def to_rating(ml_latest_row):
    user, movie, rating, timestamp = ml_latest_row.split(',')
    return recommendation.Rating(int(user), int(movie), float(rating))

def skip_header(rdd):
    header = rdd.first()
    rdd = rdd.filter(lambda row: row != header)
    return rdd

if __name__ == '__main__':
    sc = SparkContext(appName='RecSys Non-Personalized')
    data = sc.textFile("./path/to/ratings.csv")
    data = skip_header(data)
    movielens = data.map(to_rating)
    movie_ratings = movielens.map(lambda r: (r.product, r.rating)).groupByKey()
    movie_avg_rating = movie_ratings.map(lambda (movie_id, ratings): (movie_id,
        sum(ratings) / len(ratings)))
    movie_avg_rating.map(lambda (k, v): (v, k)).sortByKey(ascending=False).take(10)
```

```
from pyspark.mLLib import recommendation
from pyspark import SparkContext

def to_rating(ml_latest_row):
    user, movie, rating, timestamp = ml_latest_row.split(',')
    return recommendation.Rating(int(user), int(movie), float(rating))

def skip_header(rdd):
    header = rdd.first()
    rdd = rdd.filter(lambda row: row != header)
    return rdd

if __name__ == '__main__':
    sc = SparkContext(appName='RecSys Non-Personalized')
    data = sc.textFile("./path/to/ratings.csv")
    data = skip_header(data)
    movielens = data.map(to_rating)
    movie_ratings = movielens.map(lambda r: (r.product, r.rating)).groupByKey()
    movie_avg_rating = movie_ratings.map(lambda (movie_id, ratings): (movie_id,
        sum(ratings) / len(ratings)))
    movie_avg_rating.map(lambda (k, v): (v, k)).sortByKey(ascending=False).take(10)
```

```
from pyspark.mllib import recommendation
from pyspark import SparkContext

def to_rating(ml_latest_row):
    user, movie, rating, timestamp = ml_latest_row.split(',')
    return recommendation.Rating(int(user), int(movie), float(rating))

def skip_header(rdd):
    header = rdd.first()
    rdd = rdd.filter(lambda row: row != header)
    return rdd

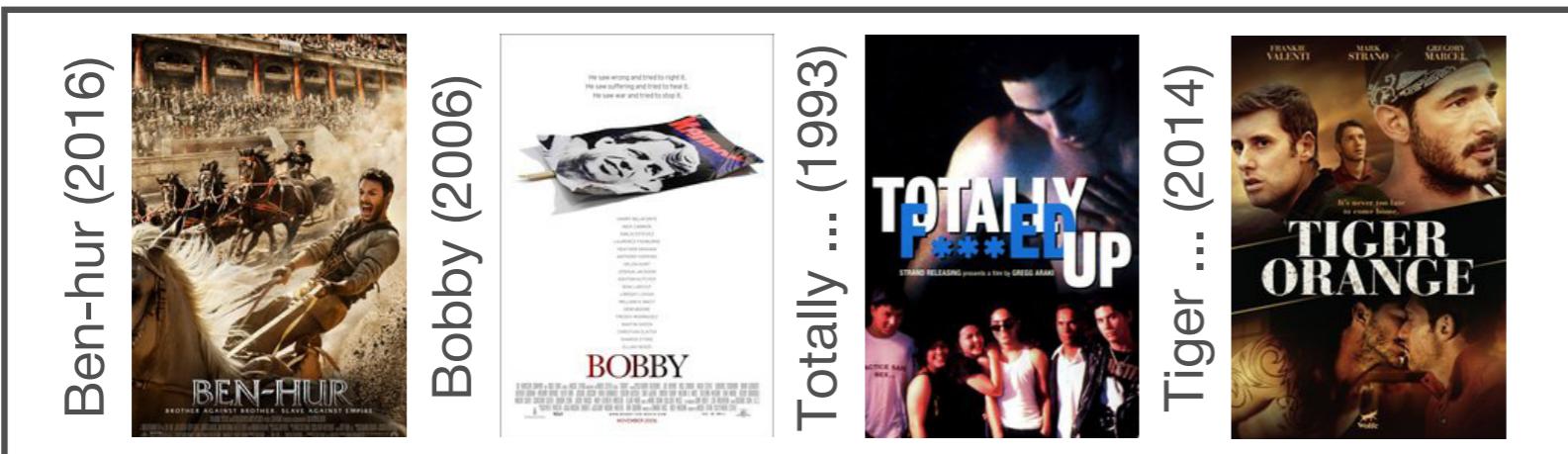
if __name__ == '__main__':
    sc = SparkContext(appName='RecSys Non-Personalized')
    data = sc.textFile("./path/to/ratings.csv")
    data = skip_header(data)
    movielens = data.map(to_rating)
    movie_ratings = movielens.map(lambda r: (r.product, r.rating)).groupByKey()
    movie_avg_rating = movie_ratings.map(lambda (movie_id, ratings): (movie_id,
        sum(ratings) / len(ratings)))
    movie_avg_rating.map(lambda (k, v): (v, k)).sortByKey(ascending=False).take(10)
```

```

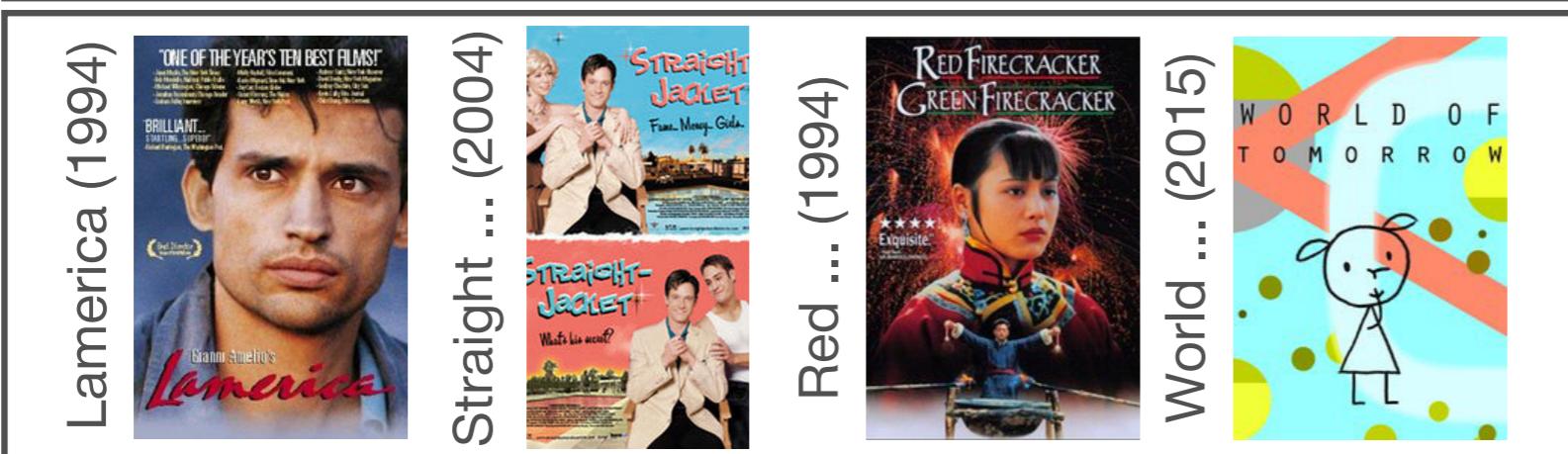
movie_titles = sc.textFile("./ml-latest-small/movies.csv")
movie_titles = skip_header(movie_titles)
titles = movie_titles.map(parse_movie_data)
# call several times to see random behaviour
print movie_avg_rating.join(titles) \
    .map(lambda (movie_id, (avg_rating, title)): (avg_rating, (title, movie_id))) \
    .sortByKey(ascending=False).take(15)

```

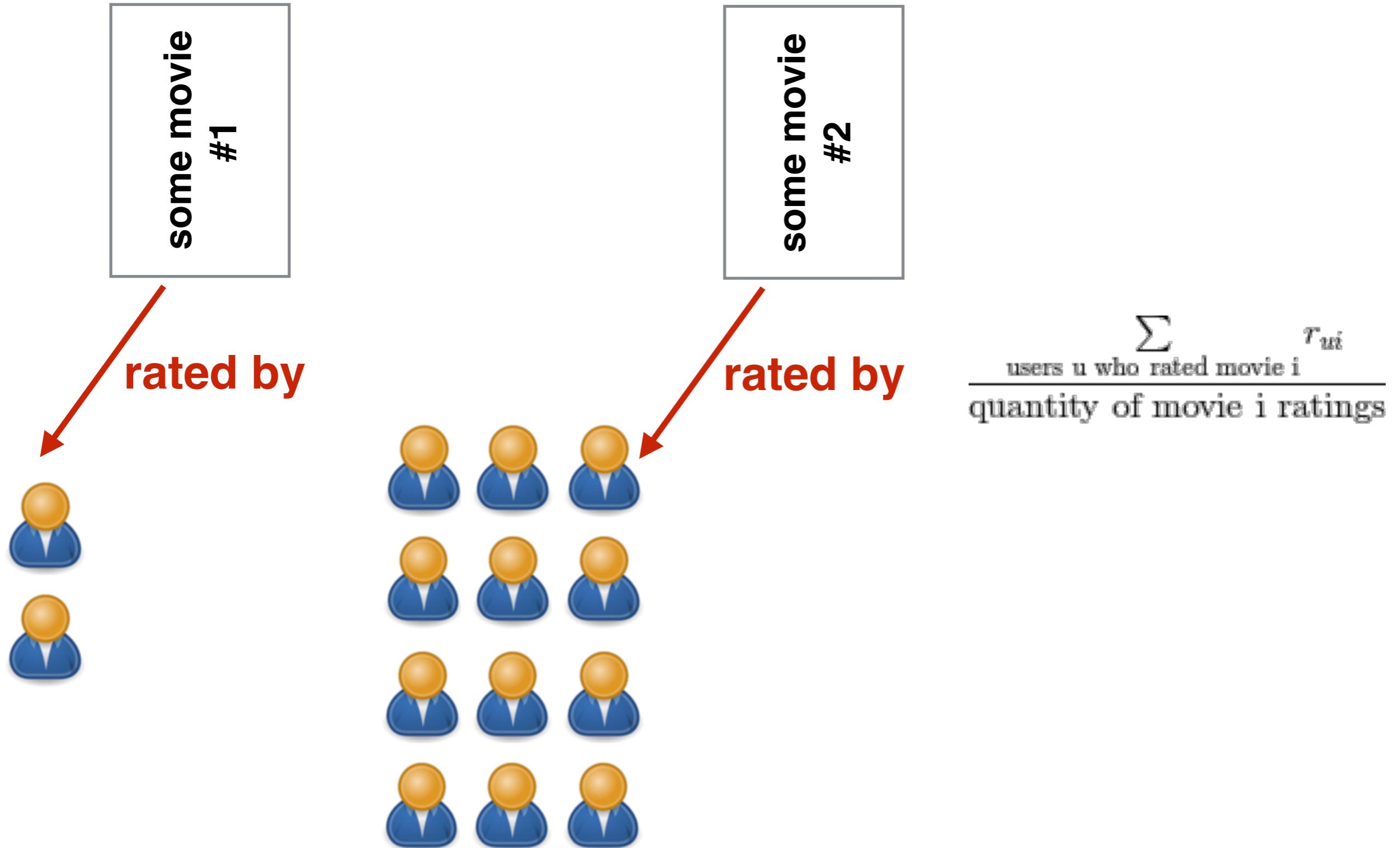
trial#1



trial#2



Damped Means



Damped Means



...

```
movie_avg_rating = movie_ratings.map(lambda (movie_id, ratings):  
(movie_id, (sum(ratings) + k * mu) / (len(ratings) + k)))
```

)

...

trial#1



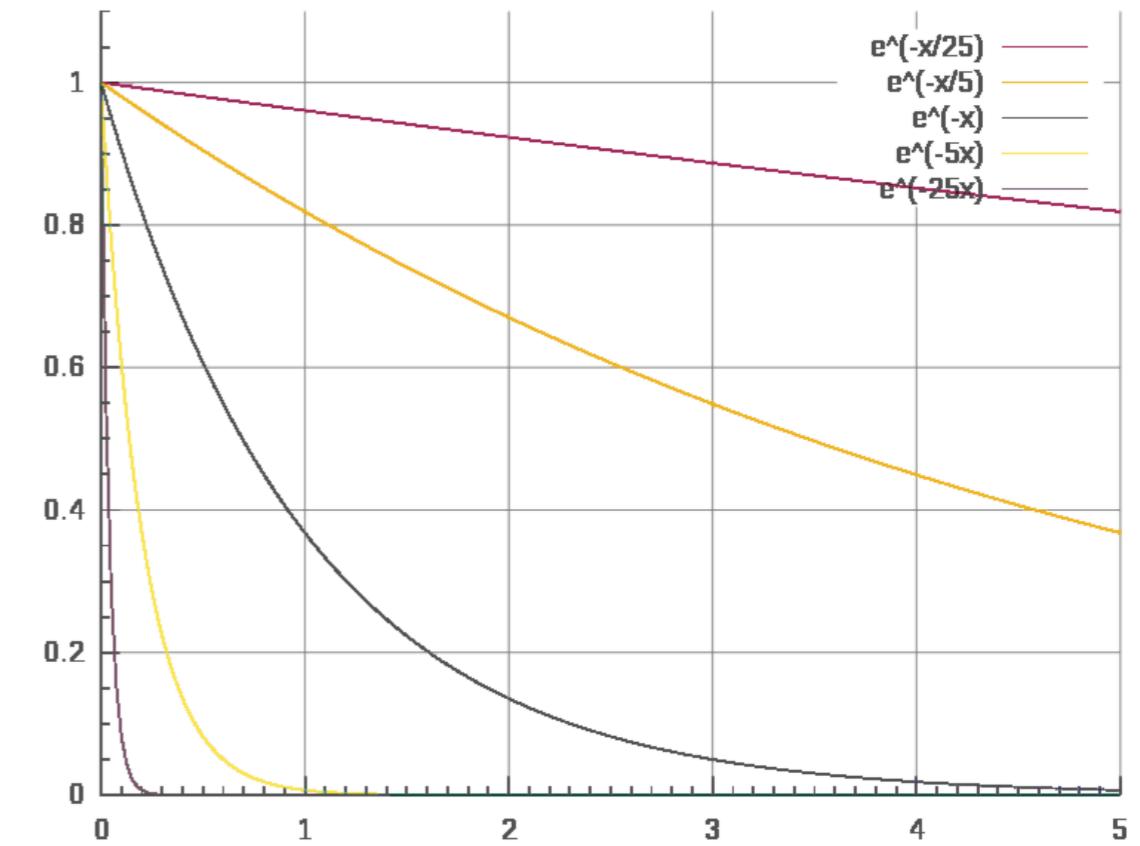
trial#2



movielens

```
$ head ~/path/to/ratings.csv
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
...
```

Trending now

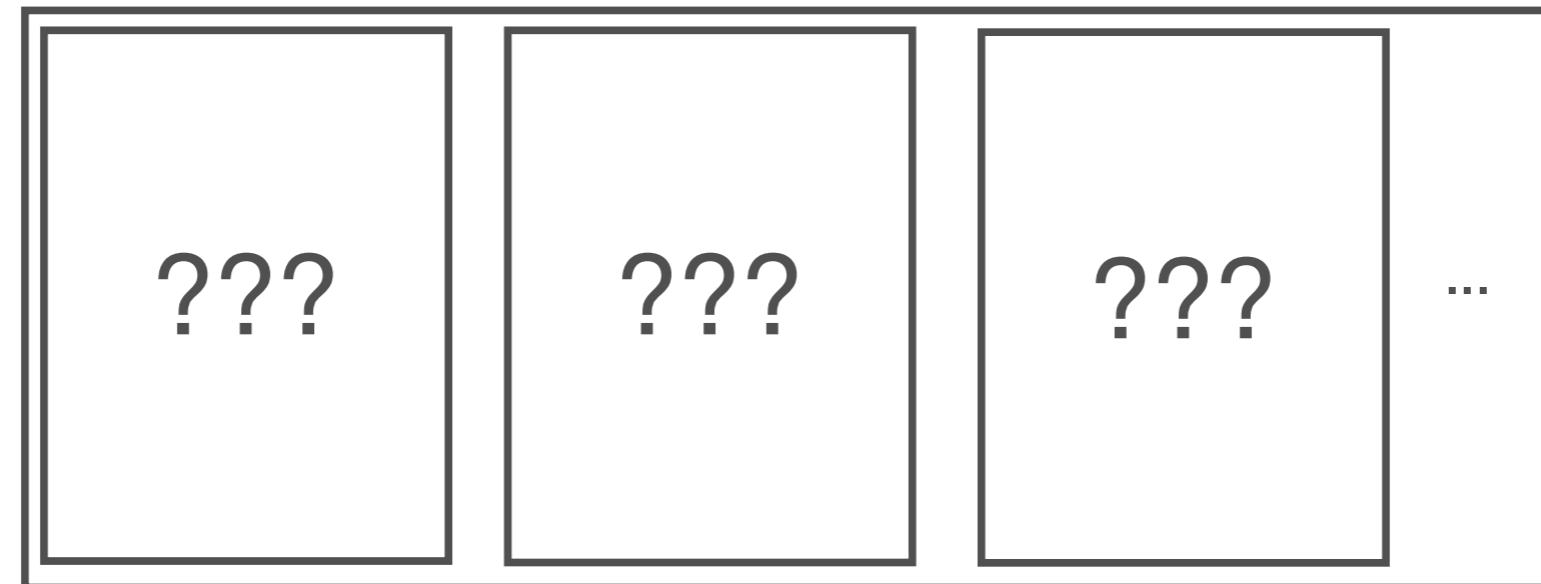


see: https://en.wikipedia.org/wiki/Exponential_decay

Trending now



$$decay = e^{-\alpha(now() - rating \text{ timestamp})}$$



Are movies getting better with time?

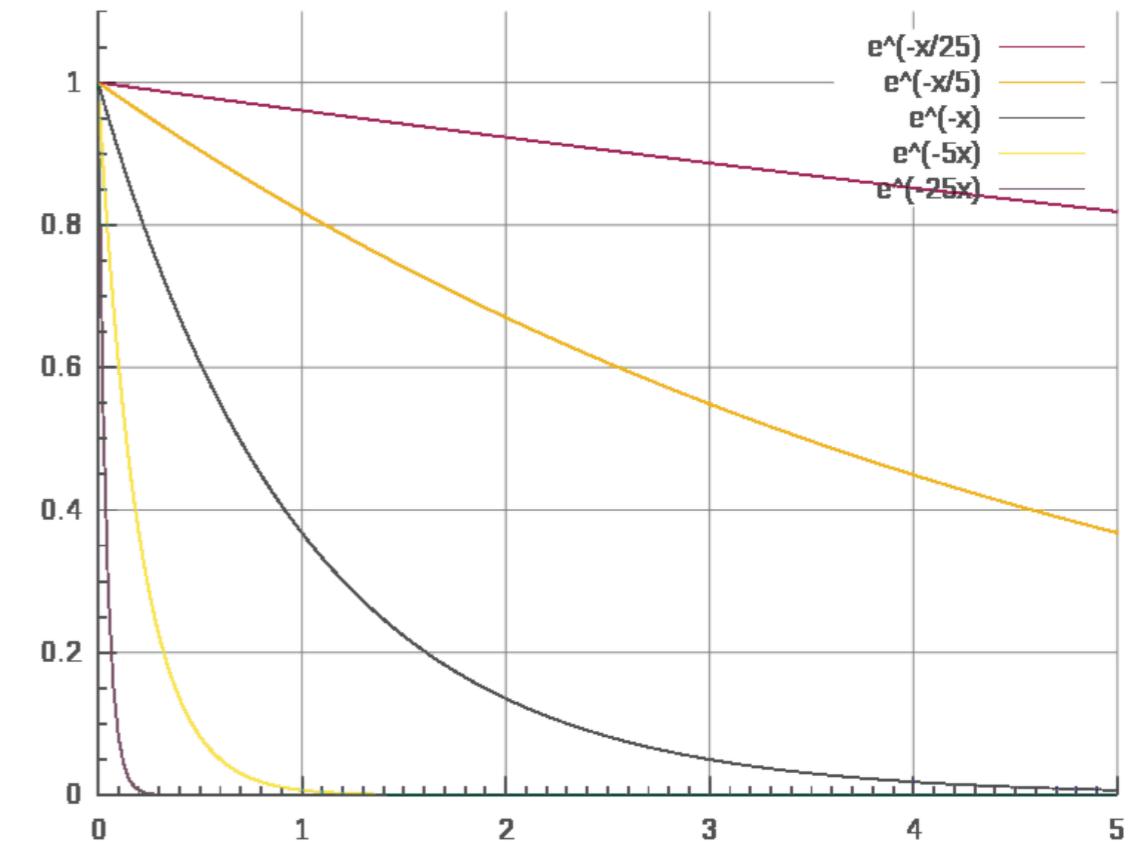


credit to: Alex Smola



Carnegie Mellon University

Trending now



see: https://en.wikipedia.org/wiki/Exponential_decay

Something Happened in Early 2004...

Netflix ratings by date



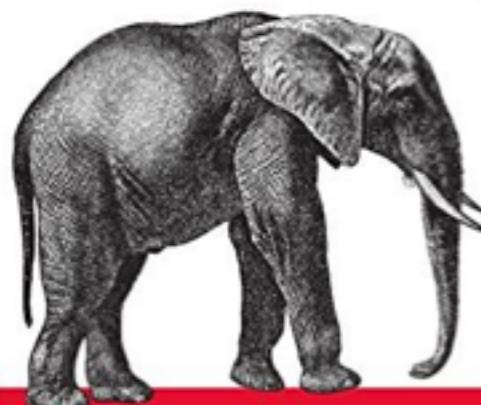
credit to: Alex Smola



rsity

O'REILLY®

4th Edition
Revised & Updated



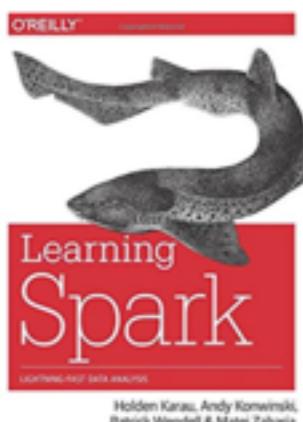
Hadoop

The Definitive Guide

STORAGE AND ANALYSIS AT INTERNET SCALE

Tom White

Customers who bought this item also bought



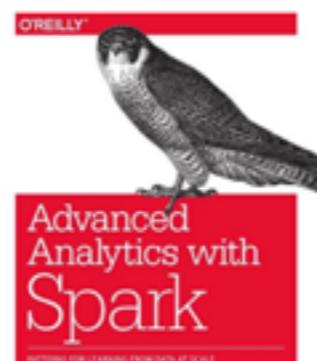
Learning Spark: Lightning-Fast Big Data Analysis

› Holden Karau

★★★★★ 58

Paperback

\$31.61 



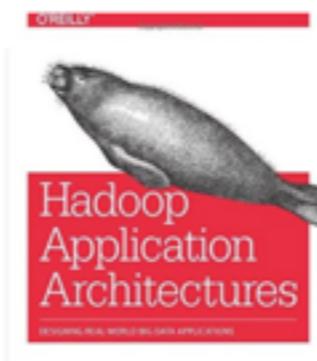
Advanced Analytics with Spark: Patterns for Learning from Data at Scale

Sandy Ryza

★★★★★ 22

Paperback

\$28.99 



Hadoop Application Architectures: Designing Real-World Big Data Applications

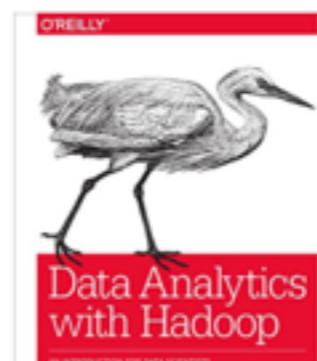
Mark Grover, Ted Malaska,
Jonathan Seidman & Gwen Shapira

› Mark Grover

★★★★★ 8

Paperback

\$36.74 



Data Analytics with Hadoop: An Introduction for Data Scientists

Benjamin Bengfort & Jenny Kim

› Benjamin Bengfort

★★★★★ 1

Paperback

\$20.30 



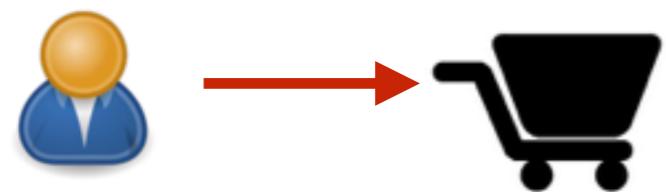
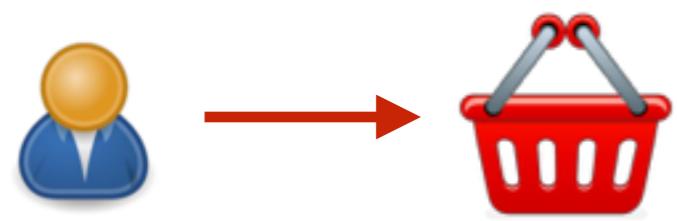
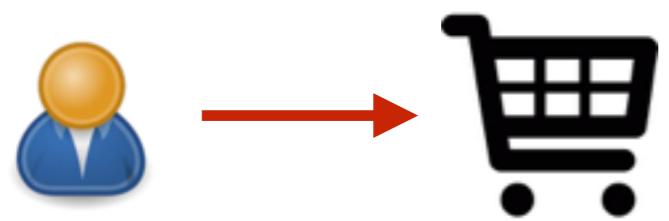
Programming Hive: Data Warehouse and Query Language for Hadoop

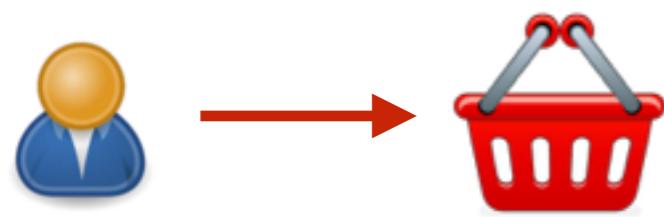
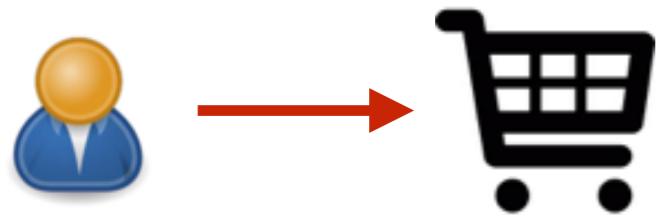
› Edward Capriolo

★★★★★ 15

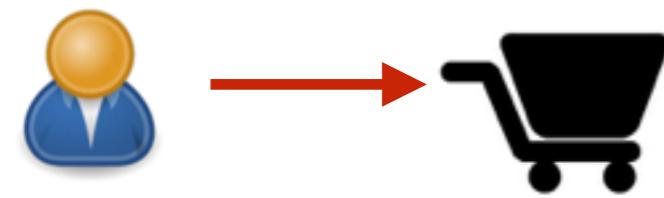
Paperback

\$33.78 

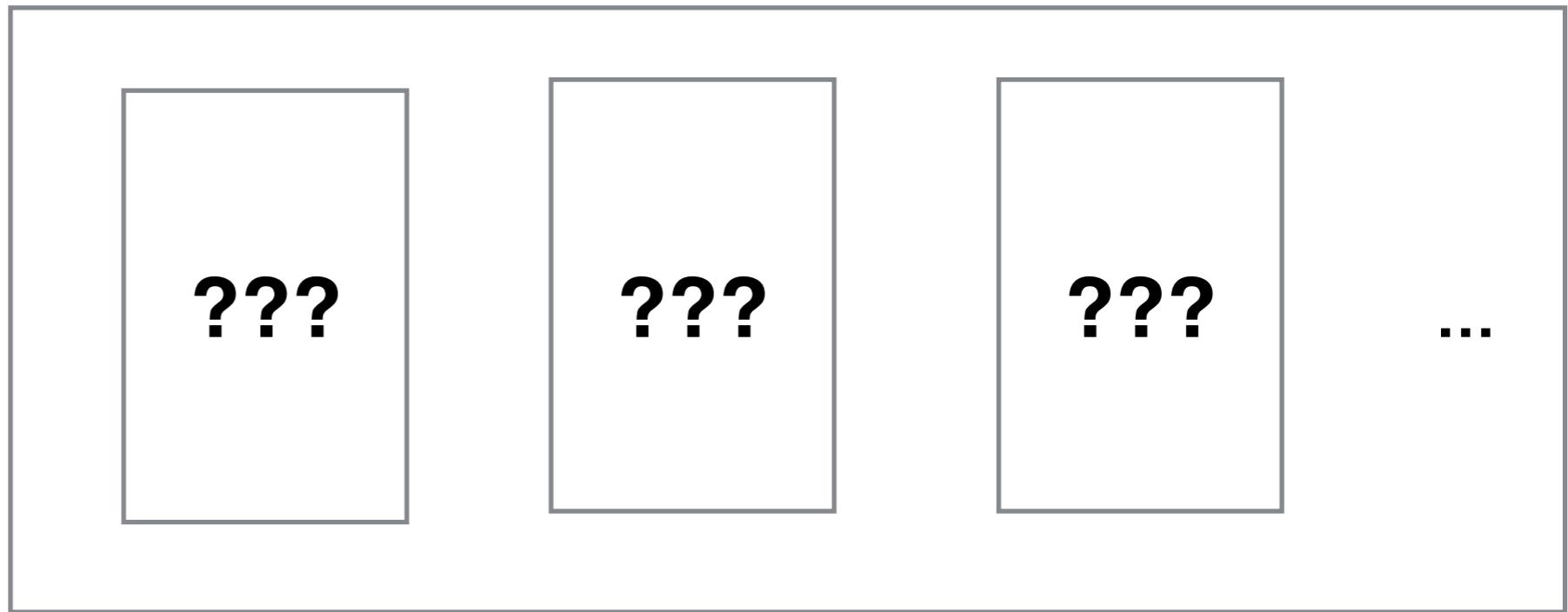




...

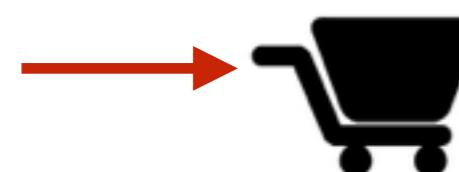


...





...



...

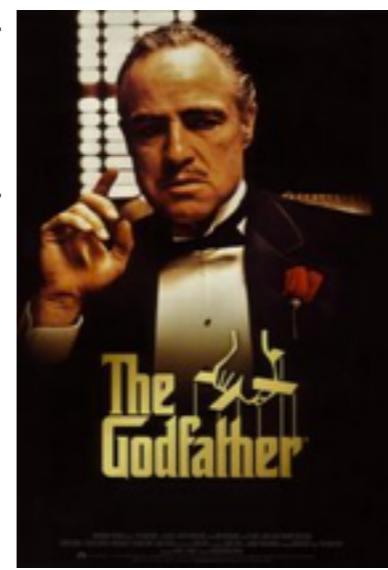
item A



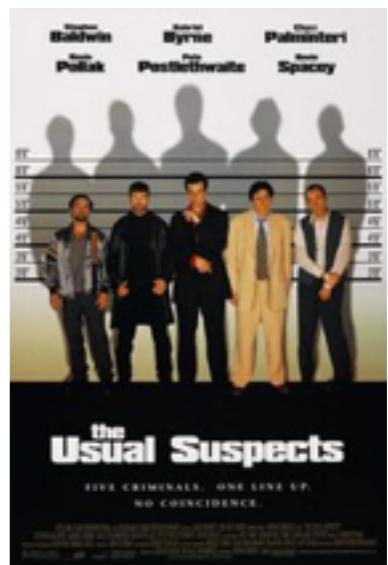
Shawshank ... (1994)



Godfather (1972)



... Suspects (1995)



Solving «Bananas» Problem

item A



movie «bananas»



Solving «Bananas» Problem

item A



$$\frac{\#A \& B}{\#A}$$



$$lift = \frac{\#A \& B}{\#A \times \#B}$$

movie «bananas»



Solving «Bananas» Problem

item A



$$\frac{\#A \& B}{\#A}$$



$$lift = \frac{\#A \& B}{\#A \times \#B}$$

movie «bananas»



$$\frac{P(B|A)}{P(B|not A)}$$

Solving «Bananas» Problem

item A



$$\frac{\#A \& B}{\#A}$$



$$lift = \frac{\#A \& B}{\#A \times \#B}$$

movie «bananas»



$$\frac{P(B|A)}{P(B|not A)}$$

see (for example): Data-Intensive Text Processing with MapReduce by Jimmy Lin and Chris Dyer. Morgan & Claypool Publishers, 2010. Chapter 3: “Pairs vs Stripes”

Coffee and Tea Break



What is the name of the following problem?

- recommend the most popular items disregard the user's context / behavior / history.

Minions Problem

Bananas Problem

Winnie the Pooh Problem

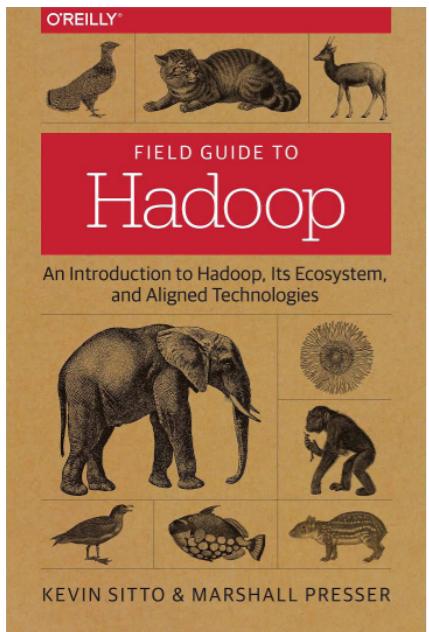
Harry Potter Problem

Orange Problem

Outline (Introduction into RS)

- RS Basics: domains, classification
- Non-personalised RS
- **Content-Based RS**
- RS Evaluation

Content Based (CB)



Property	Expected Type	Description
Properties from Book		
bookEdition	Text	The edition of the book.
bookFormat	BookFormatType	The format of the book.
illustrator	Person	The illustrator of the book.
isbn	Text	The ISBN of the book.
numberOfPages	Integer	The number of pages in the book.

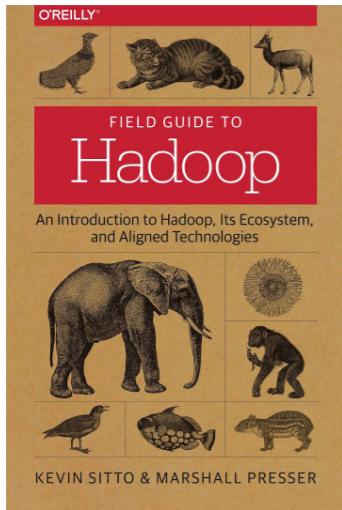
User	Book	Author	...	Target
programmer A	C++	Bjarne Stroustrup	...	likes
programmer A	Cooking	Gordon Ramsay	...	dislikes
programmer B	Cooking	Gordon Ramsay	...	likes
...
programmer A	Linux Cookbook	Carla Schroder	...	???

<http://schema.org/Book>

Questions:

1. Item representation
2. User representation
3. Distance / similarity measure

item



1



$$i \in \mathbb{R}^m$$



2

user



$$u \in \mathbb{R}^m$$



3

$$distance(u, i) : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$$

Questions:

1. Item representation
2. User representation
3. Distance / similarity measure

Movie	Budget (numerical)	Genre (categorical)	...
The Shawshank Redemption	\$25,000,000	Crime Drama	...
The Godfather	\$6,000,000	Crime Drama	...
...

Questions:

1. Item representation
2. User representation
3. Distance / similarity measure

Movie	User Reviews	...
The Shawshank Redemption	... This is simply one of the best films ever made and
The Godfather	... The Godfather is commonly considered to be one of the “greatest films of all time”	...
...

Questions:

1. Item representation
2. User representation
3. Distance / similarity measure

Text corpus:

... this is simply one of the best films ever made and ...



see Spark: Word2Vec

word



$$\in \mathbb{R}^k$$

OPEN

<https://issues.apache.org/jira/browse/SPARK-14864> [doc2vec]

BigData Notes

Questions:

1. Item representation
2. **User representation**
3. Distance / similarity measure

$$U \begin{bmatrix} \dots & 5 & 2 & 4 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & i_1 & i_2 & i_3 & \dots \end{bmatrix} \xrightarrow{\hspace{1cm}} U = AVG \left(\begin{array}{c} 5 \times i_1 \quad \boxed{} \\ 2 \times i_2 \quad \boxed{} \\ 4 \times i_3 \quad \boxed{} \end{array} \right)$$

Questions:

1. Item representation
2. **User representation**
3. Distance / similarity measure

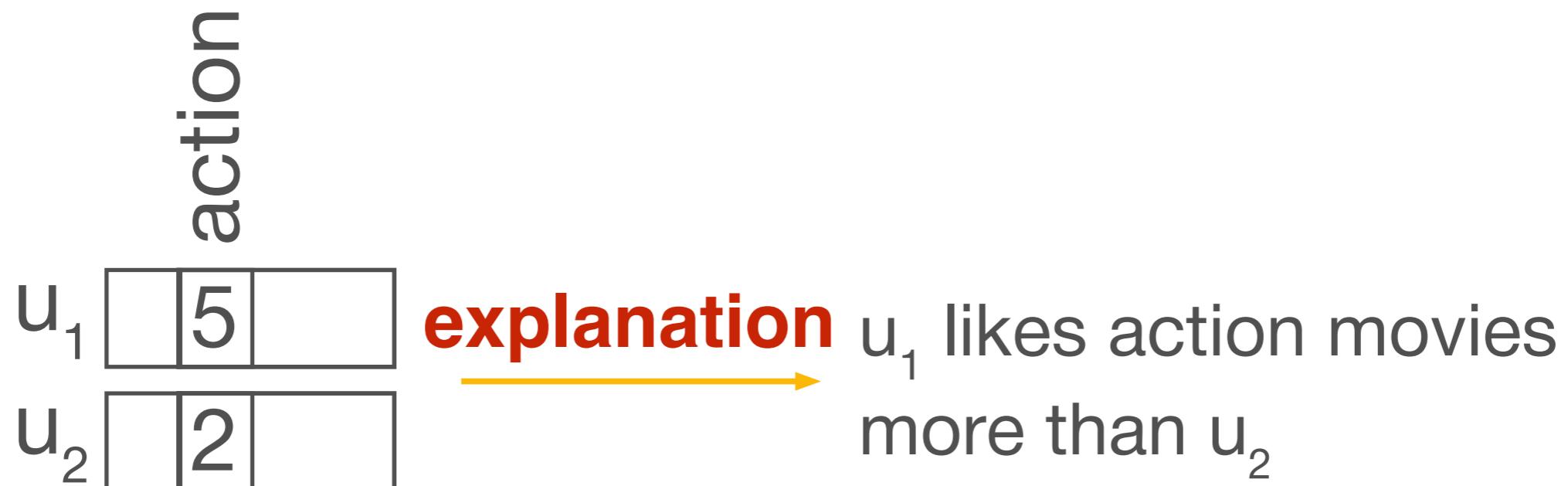
$$U \begin{bmatrix} & i_1 & i_2 & i_3 \\ & \vdots & \vdots & \vdots \\ \dots & 5 & 2 & 4 & \dots \\ & \vdots & \vdots & \vdots & \end{bmatrix} \xrightarrow{\text{---}} U = AVG \left(\begin{array}{c} 5 \times i_1 \quad \boxed{} \\ 2 \times i_2 \quad \boxed{} \\ 4 \times i_3 \quad \boxed{} \end{array} \right)$$

$$AVG : \frac{i_1 + i_2 + i_3}{3} \quad AVG : \frac{5i_1 + 2i_2 + 4i_3}{5 + 2 + 4}$$

$$AVG : \frac{decay(ts_1) \times i_1 + \dots}{decay(ts_1) + \dots}$$

Questions:

1. Item representation
2. **User representation**
3. Distance / similarity measure



Questions:

1. Item representation
2. **User representation (vice versa)**
3. Distance / similarity measure

The diagram illustrates the concept of user representation. On the left, two vectors are shown: u_1 and u_2 . Vector u_1 has elements 5, 4, and 3, while vector u_2 has elements 4, 3, and 2. An arrow points from these vectors to a central column vector i , which has elements 5, 4, 3, and 2. This central vector i is labeled as the average of u_1 and u_2 , represented by the formula $i = AVG \left(\begin{matrix} 5 \times u_1 \\ 4 \times u_2 \end{matrix} \right)$.

$$u_1 = \begin{bmatrix} \vdots \\ 5 \\ \vdots \\ 4 \end{bmatrix}, \quad u_2 = \begin{bmatrix} \vdots \\ 4 \\ \vdots \\ 3 \end{bmatrix}$$
$$i = AVG \left(\begin{matrix} 5 \times u_1 \\ 4 \times u_2 \end{matrix} \right)$$

Questions:

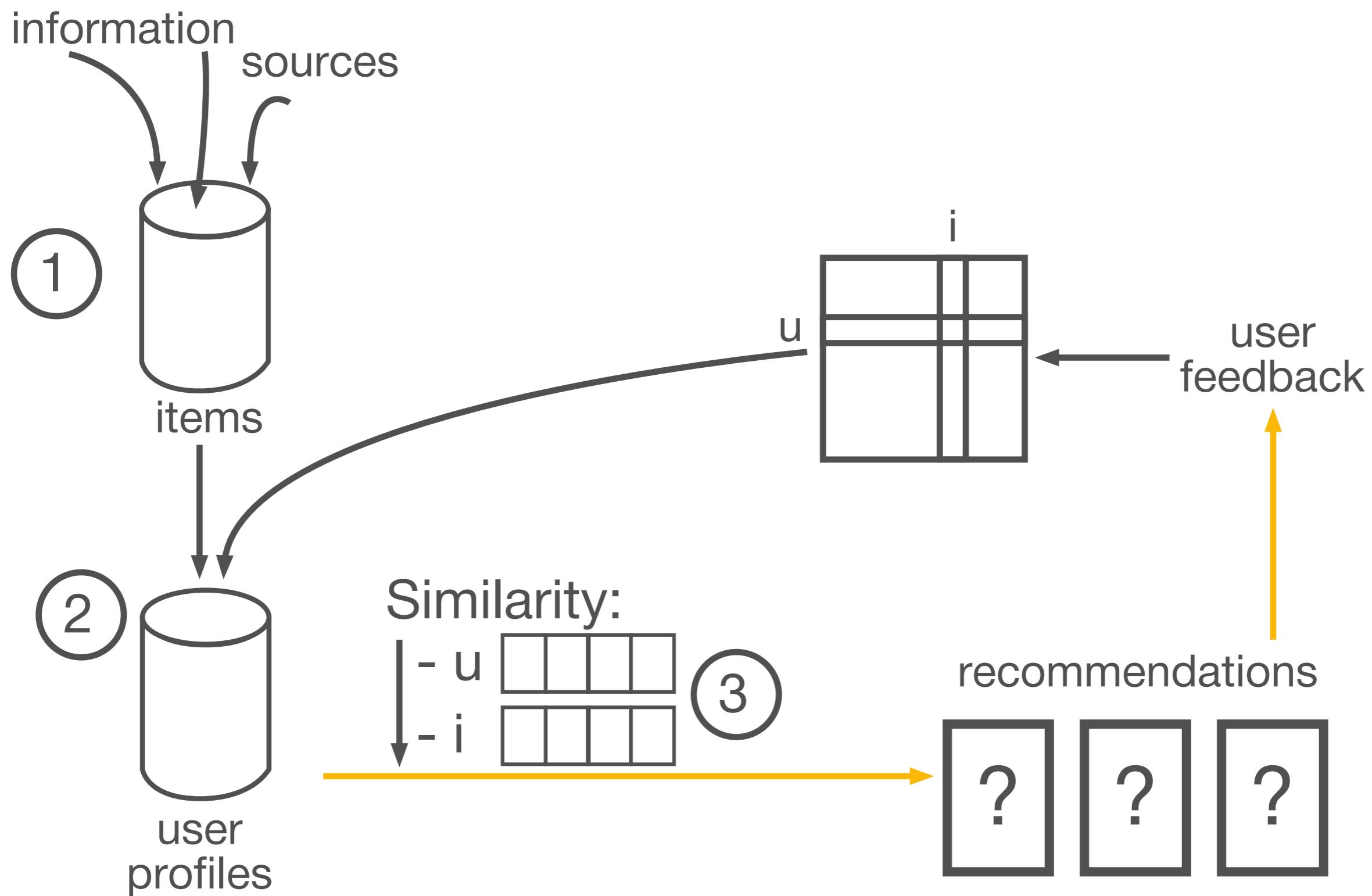
1. Item representation
2. User representation
- 3. Distance / similarity measure**



1. Metrics intended for real-valued vector spaces
2. Metrics intended for integer-valued vector spaces
3. Metrics intended for boolean-valued vector spaces
- ... user-defined distance

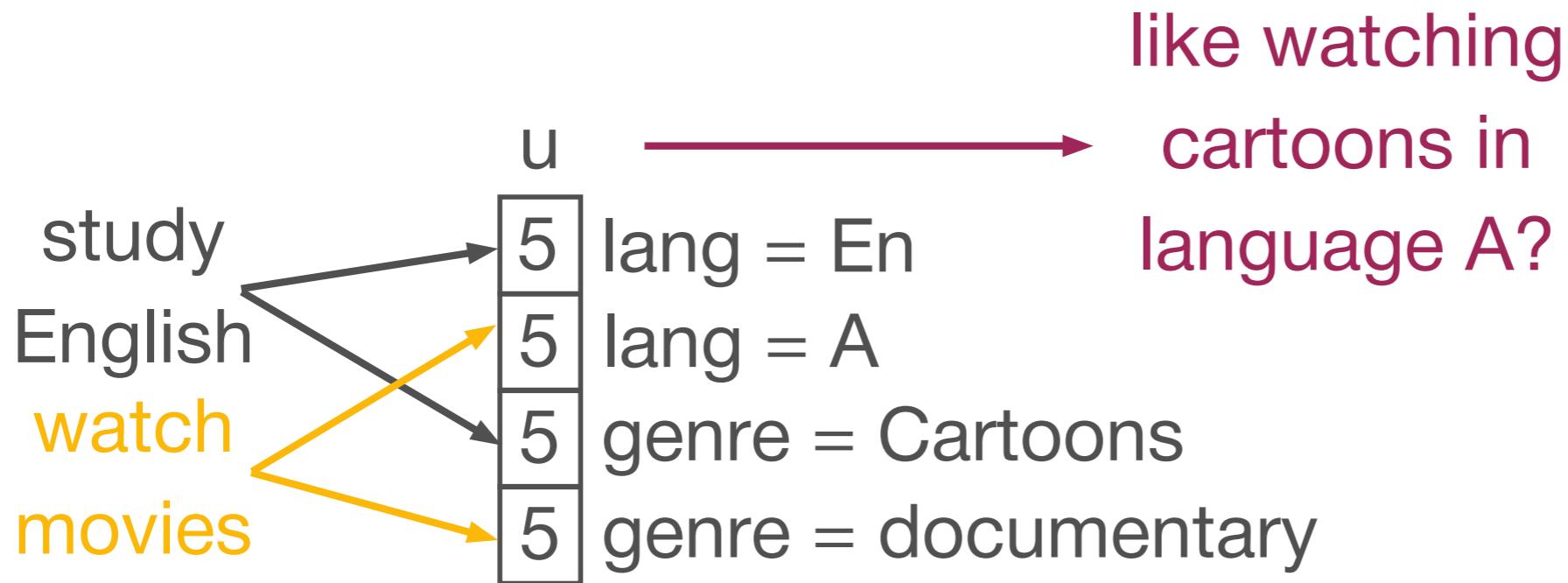
<http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html>

CB RS Cheatsheet



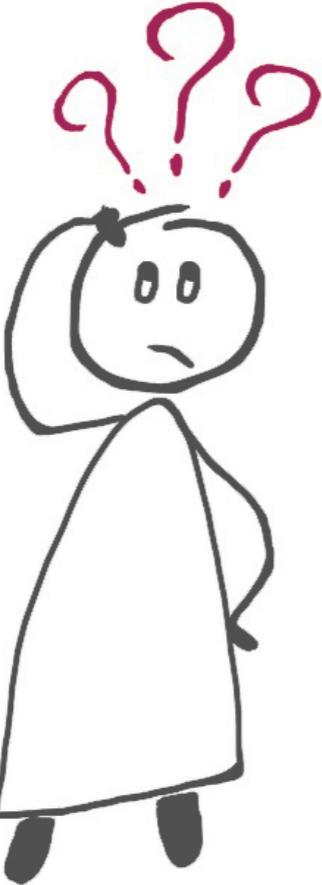
Content-Based RecSys: **Pros** and **Cons**

- (+) personalized recommendations
- (+/-) cold-start problem (either user or item)
- (+/-) profile explanations (except embeddings)
- (-) complex interdependencies

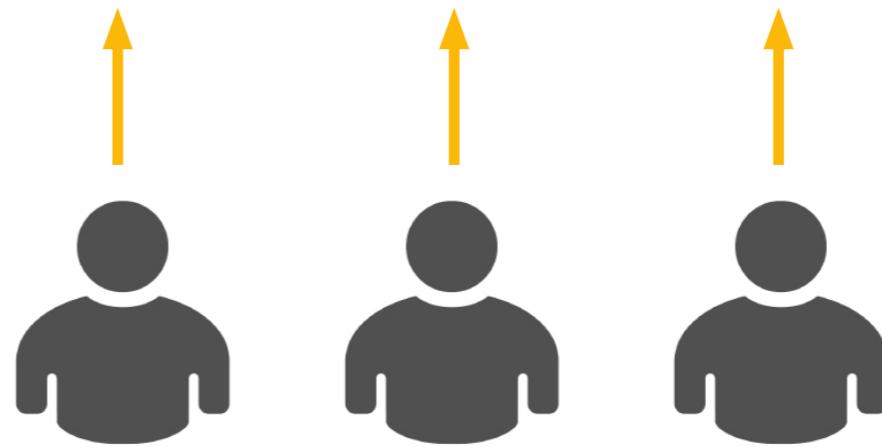


Outline (Introduction into RS)

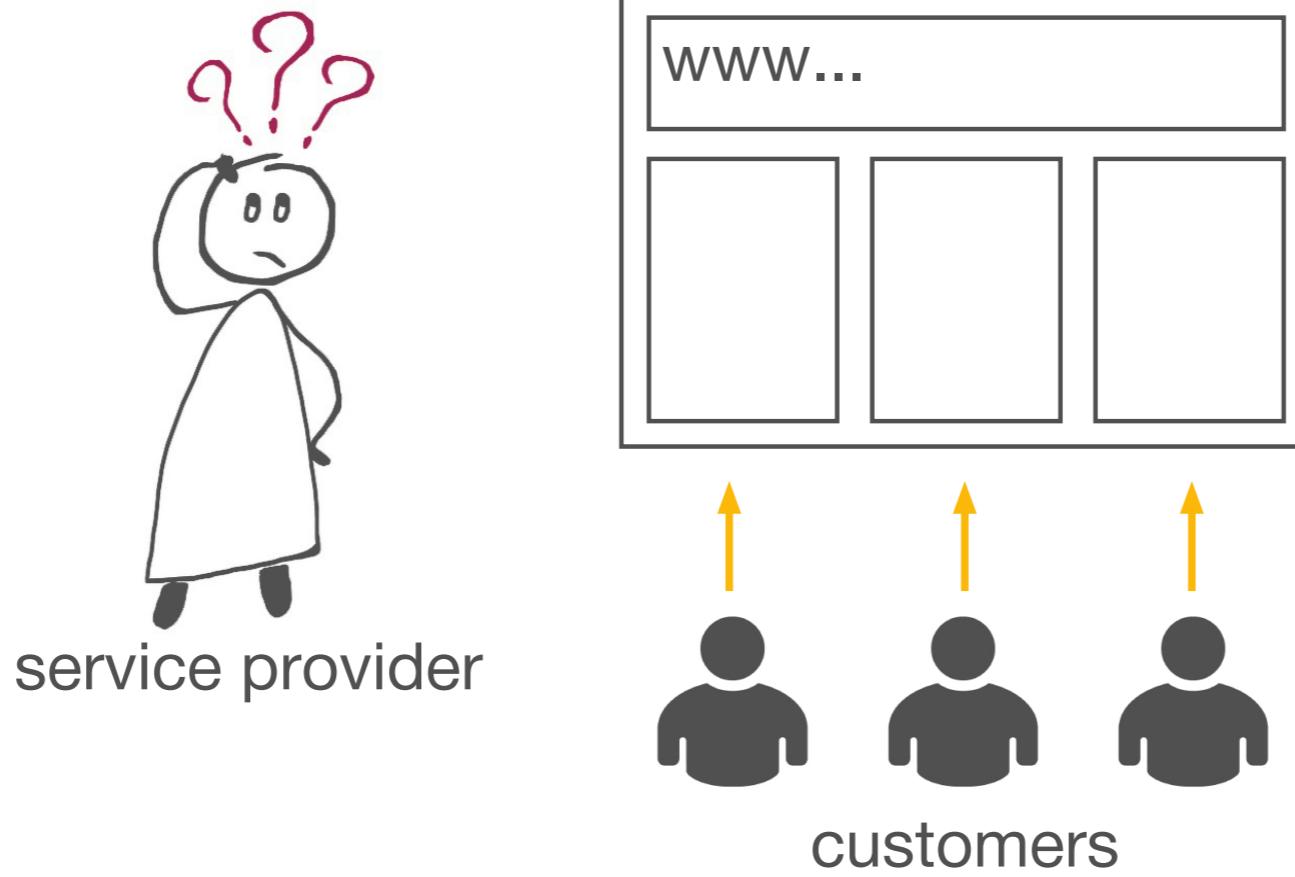
- RS Basics: domains, classification
- Non-personalised RS
- Content-Based RS
- **RS Evaluation**



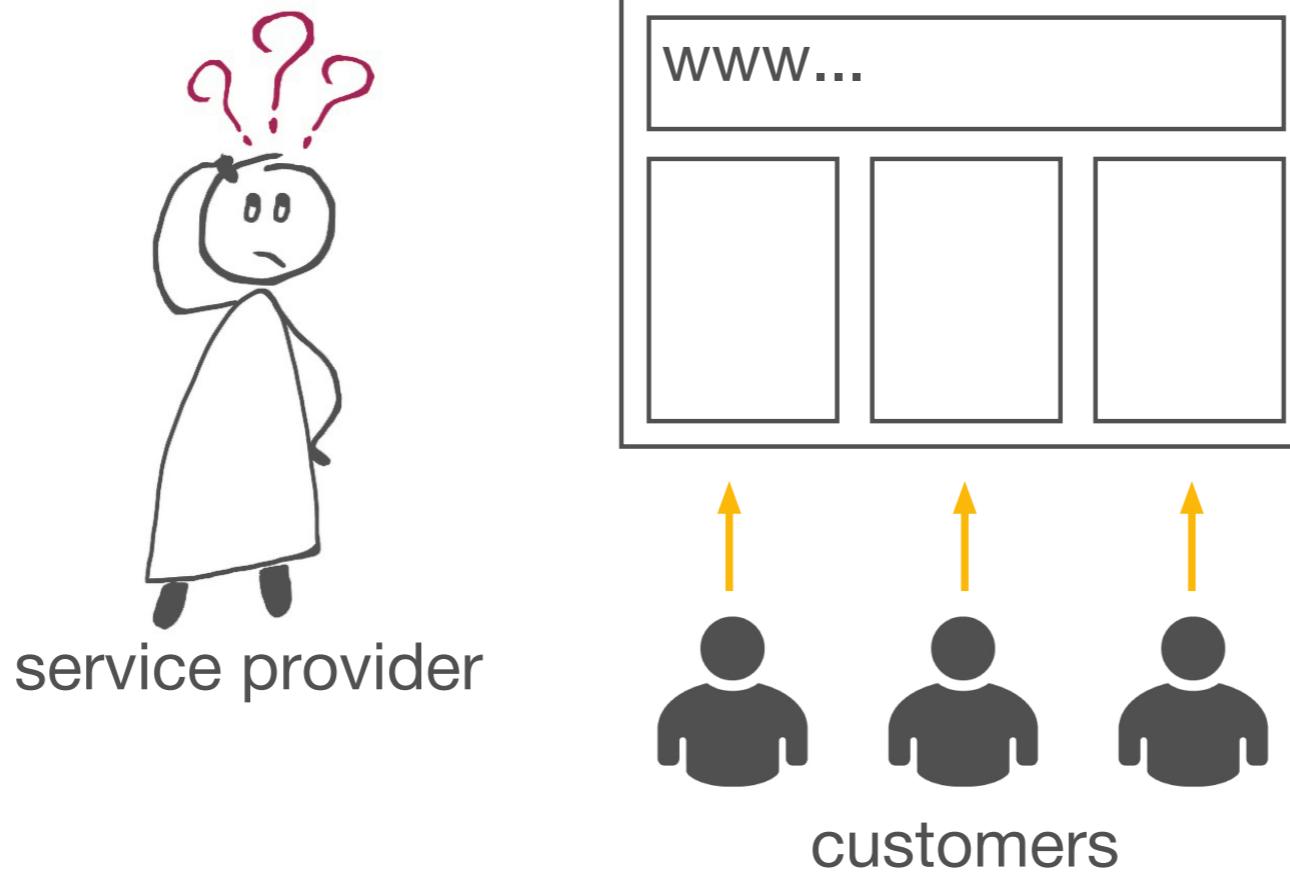
service provider



customers



long-term objective



long-term objective



$$CLV = GC \times \frac{1}{1 + d - r}$$

CLV - Customer Lifetime Value

GC - Gross Contribution

d - discount rate

r - retention rate

[пожизненная ценность клиента]



vs.





vs.



monthly LTV (**A**) vs. monthly LTV (**B**)



see: sand clock in Budapest



long term objective (e.g. LTV)

proxy

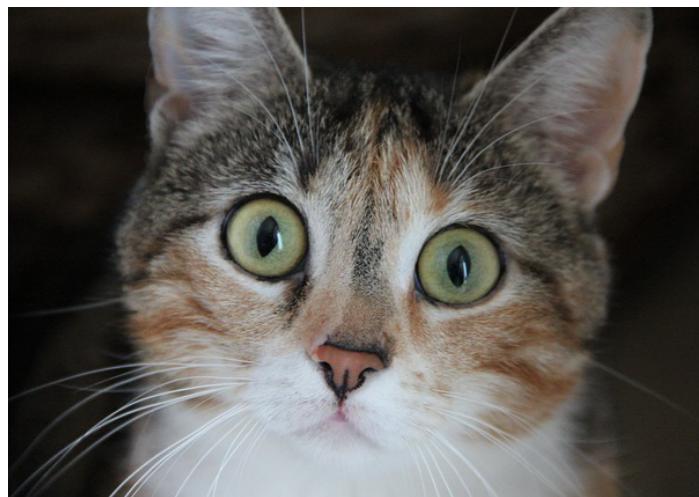
short term objective (e.g. CTR)

$$CTR = \frac{\#clicks}{\#impressions}$$

long term objective (e.g. LTV)

proxy

short term objective (e.g. CTR)



$$CTR = \frac{\#clicks}{\#impressions}$$

1. High correlation(short term objective, long term objective)

long term objective (e.g. LTV)

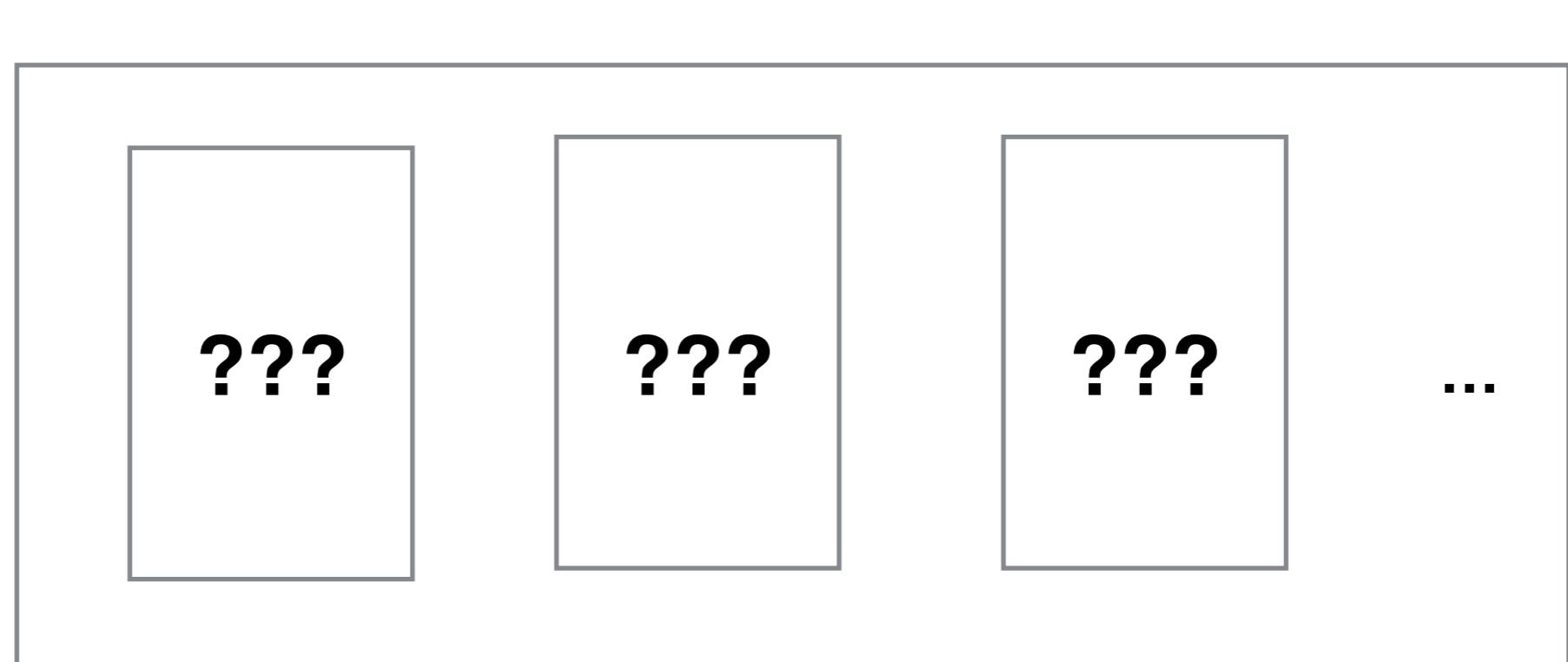
proxy

short term objective (e.g. CTR)

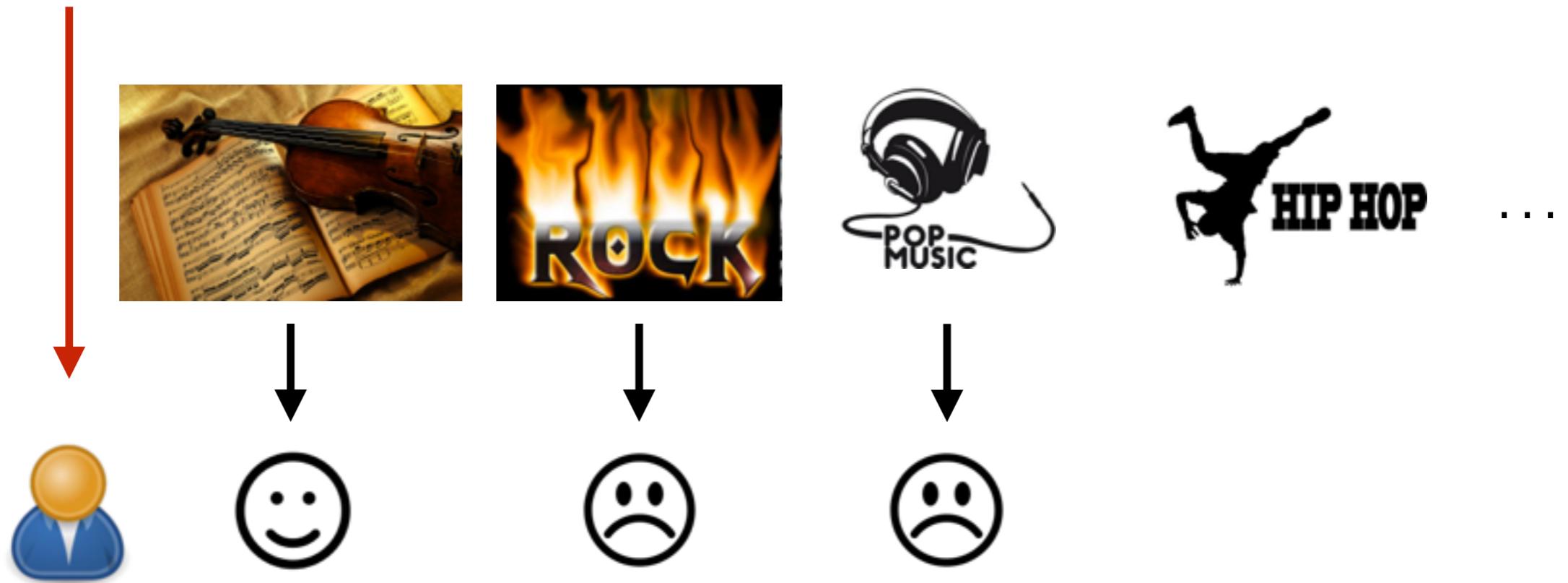
$$CTR = \frac{\#clicks}{\#impressions}$$

1. High correlation(short term objective, long term objective)
2. Faster results

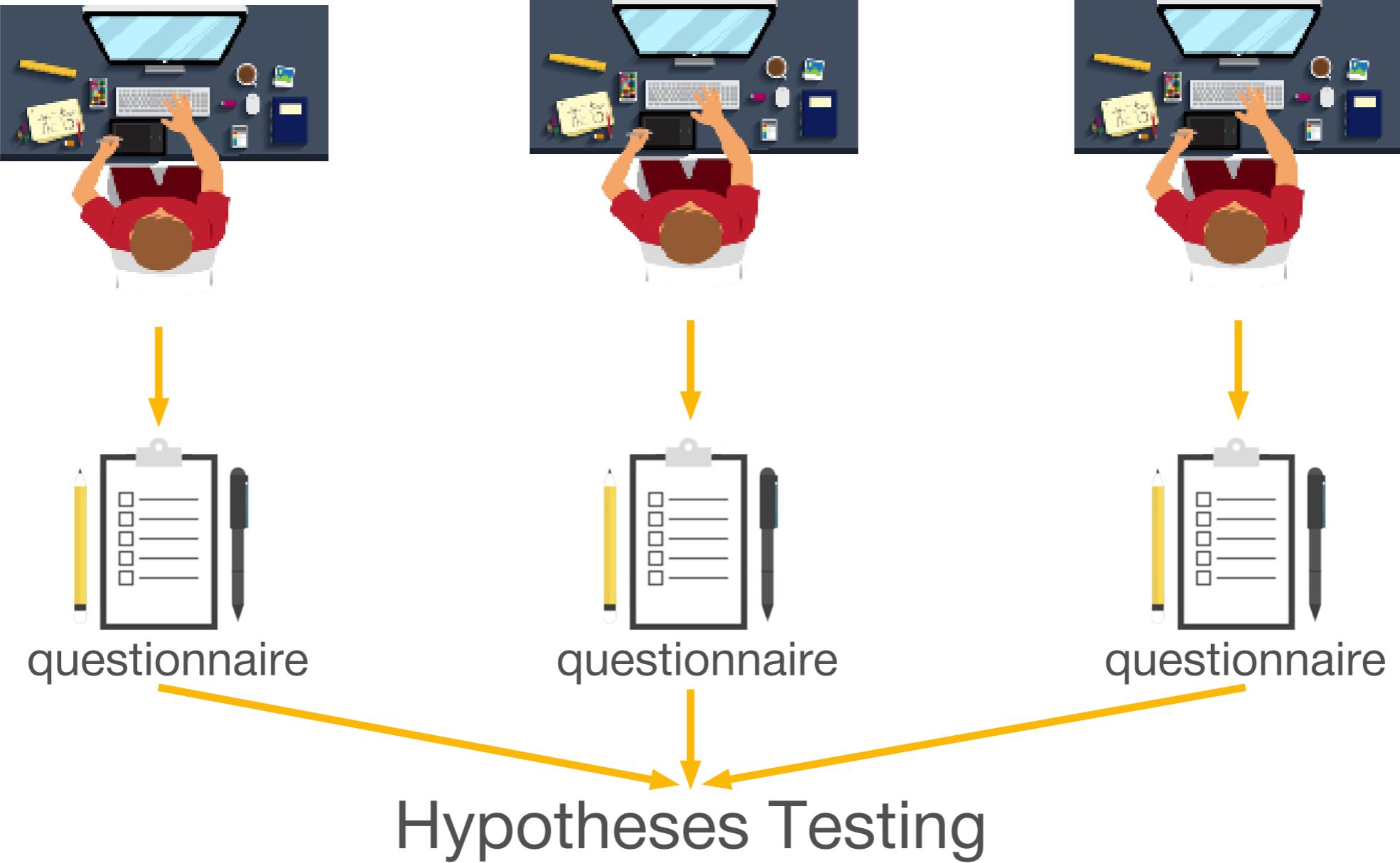
online experiments risks



online experiments risks

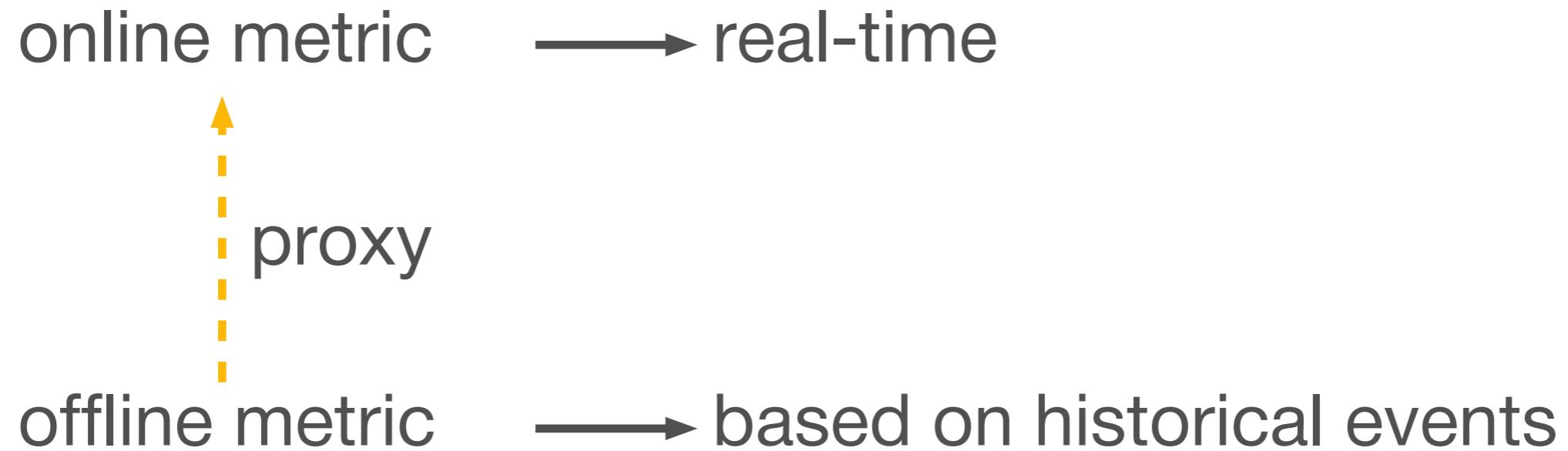


User studies



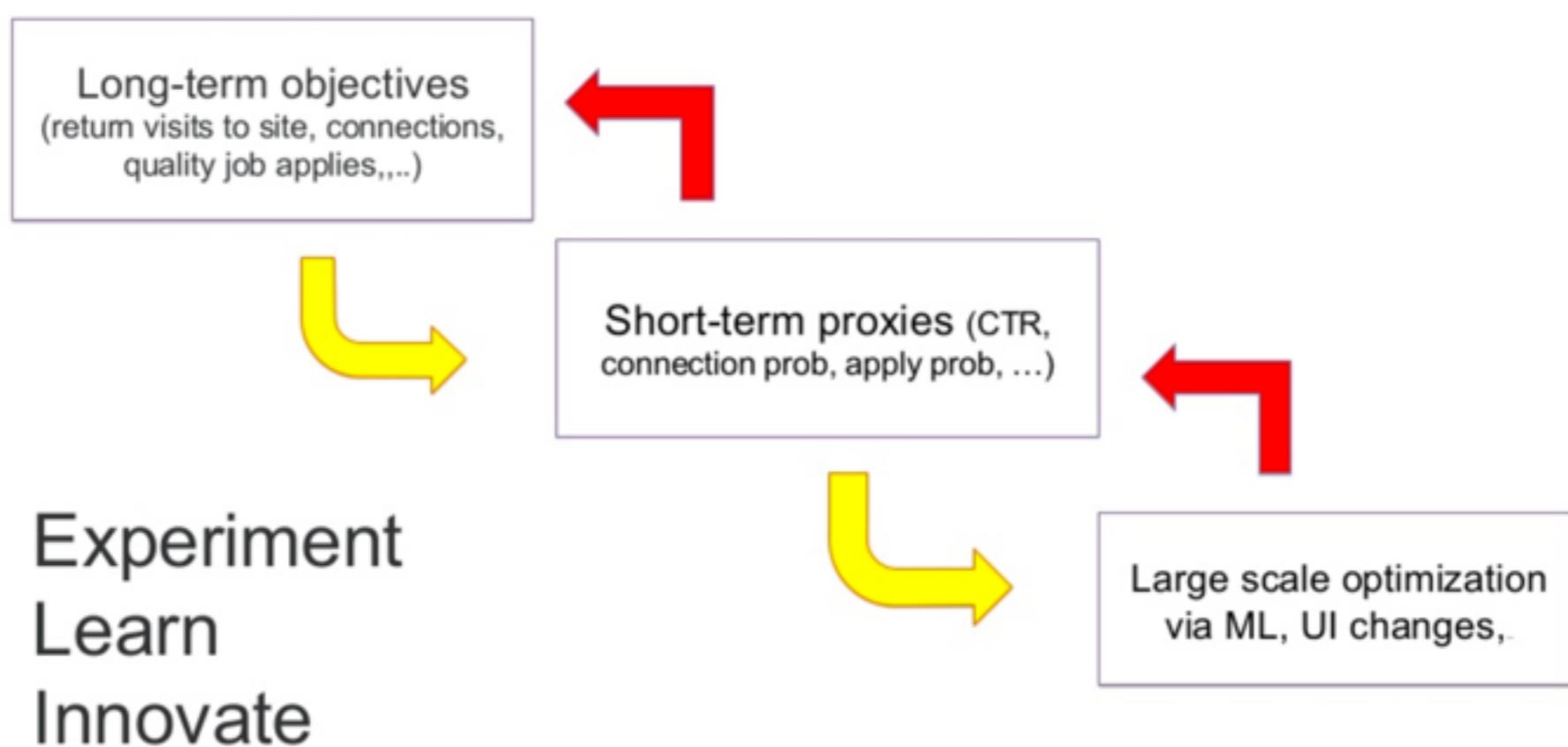
see: https://en.wikipedia.org/wiki/Statistical_hypothesis_testing

Offline experiments



RecSys'16 tutorial: Lessons learned from building real-life recommender systems

Connecting long-term objectives to proxies that can be optimized by machines/algorithms Clip slide



credit: Deepak Agarwal, Senior Director of Engineering at LinkedIn

Offline Metrics

$$MAE = \frac{\sum_{i=1}^N |p_i - r_i|}{N}$$
$$RMSE = \sqrt{\frac{\sum_{i=1}^N (p_i - r_i)^2}{N}}$$

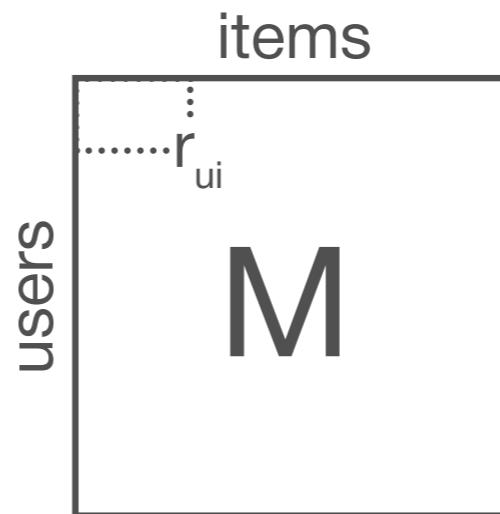
see: https://en.wikipedia.org/wiki/Mean_absolute_error

see: https://en.wikipedia.org/wiki/Root-mean-square_deviation

Offline Metrics

$$\text{GMAE} = \frac{\sum_{i=1}^N |p_i - r_i|}{N}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (p_i - r_i)^2}{N}}$$



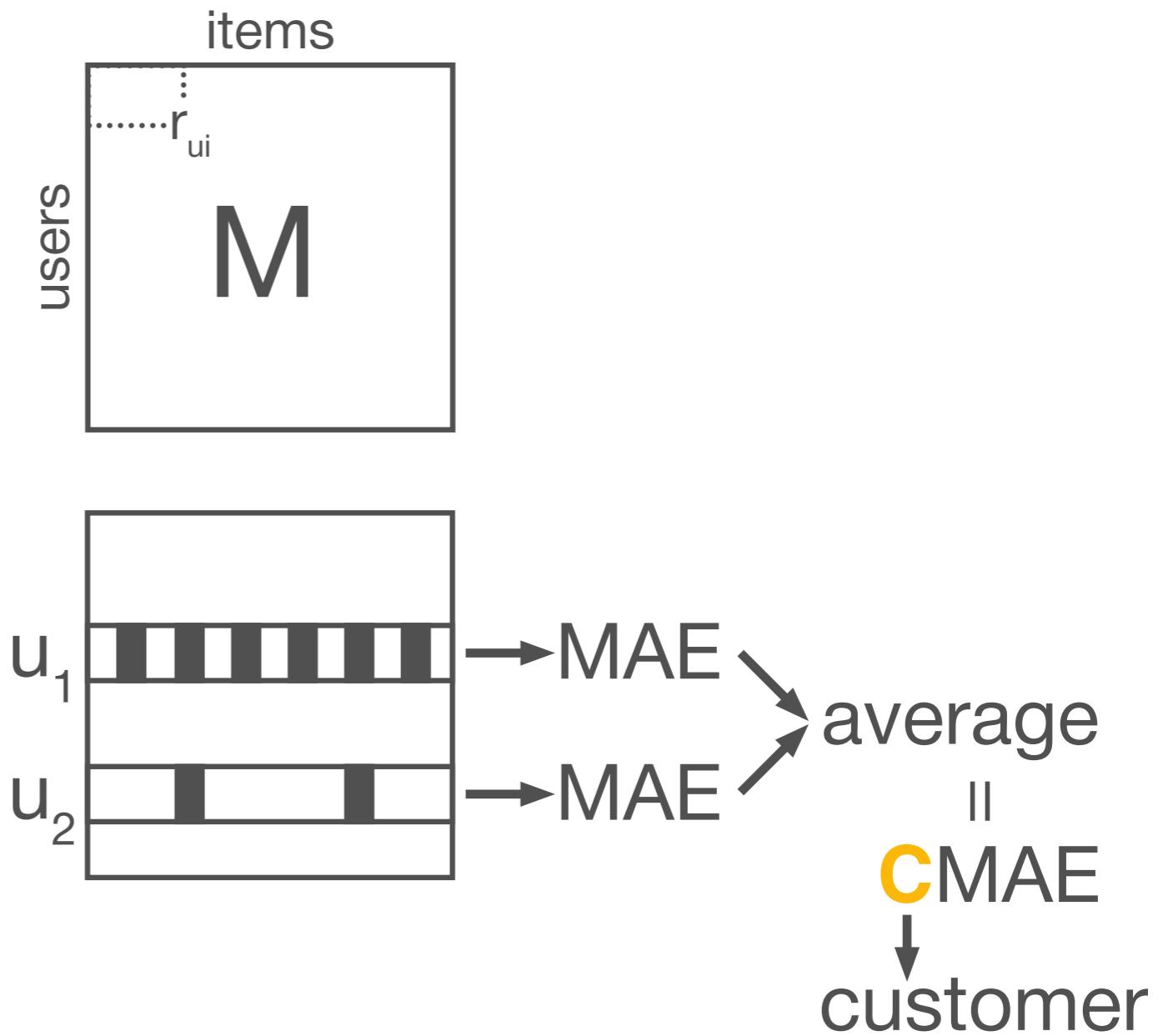
see: https://en.wikipedia.org/wiki/Mean_absolute_error

see: https://en.wikipedia.org/wiki/Root-mean-square_deviation

Offline Metrics

$$\text{GMAE} = \frac{\sum_{i=1}^N |p_i - r_i|}{N}$$

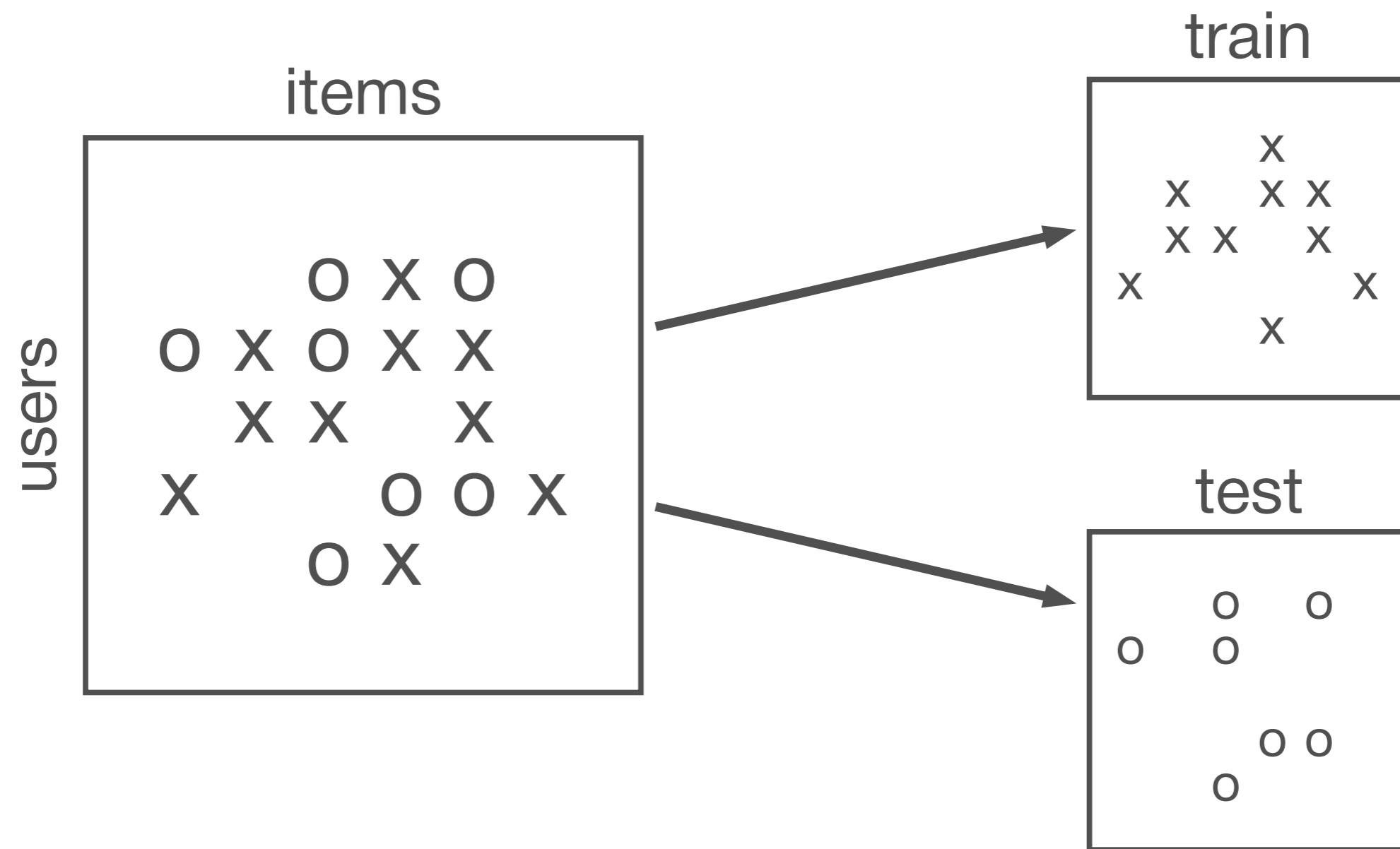
$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (p_i - r_i)^2}{N}}$$



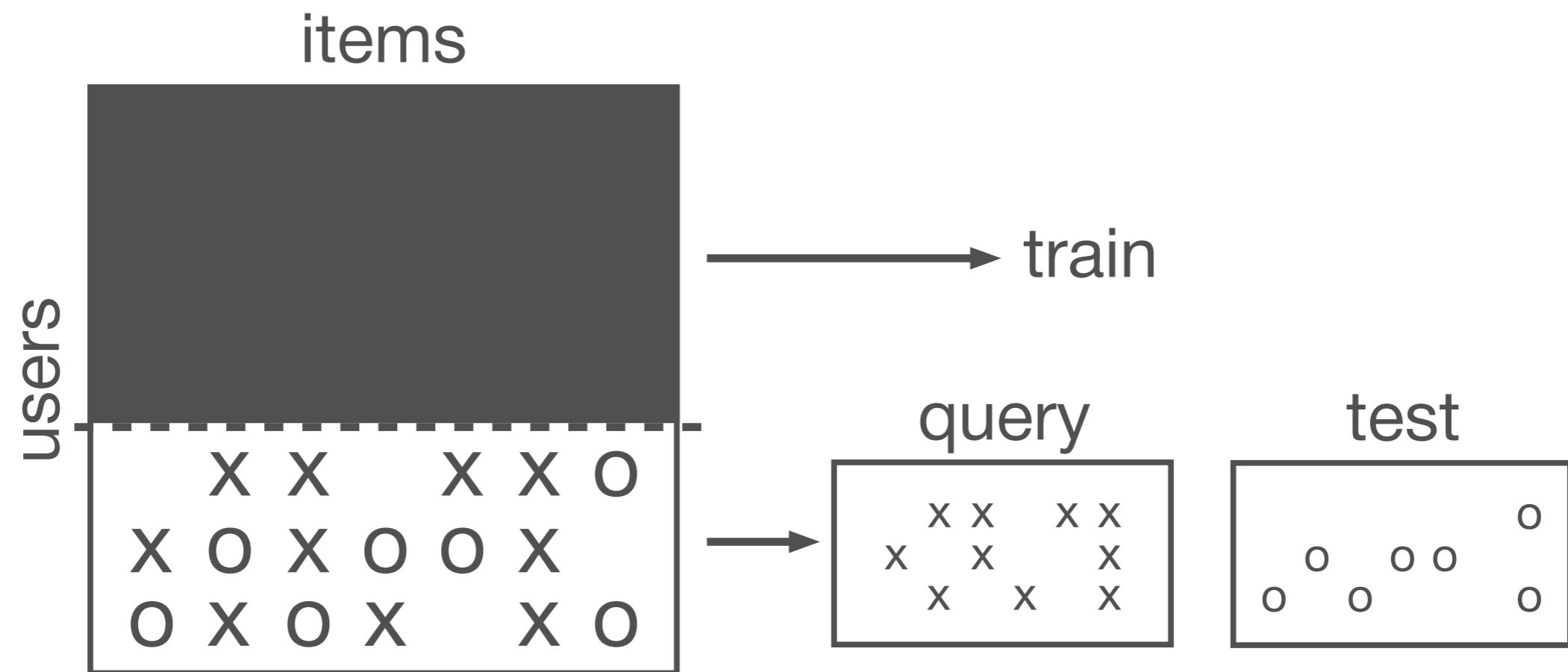
see: https://en.wikipedia.org/wiki/Mean_absolute_error

see: https://en.wikipedia.org/wiki/Root-mean-square_deviation

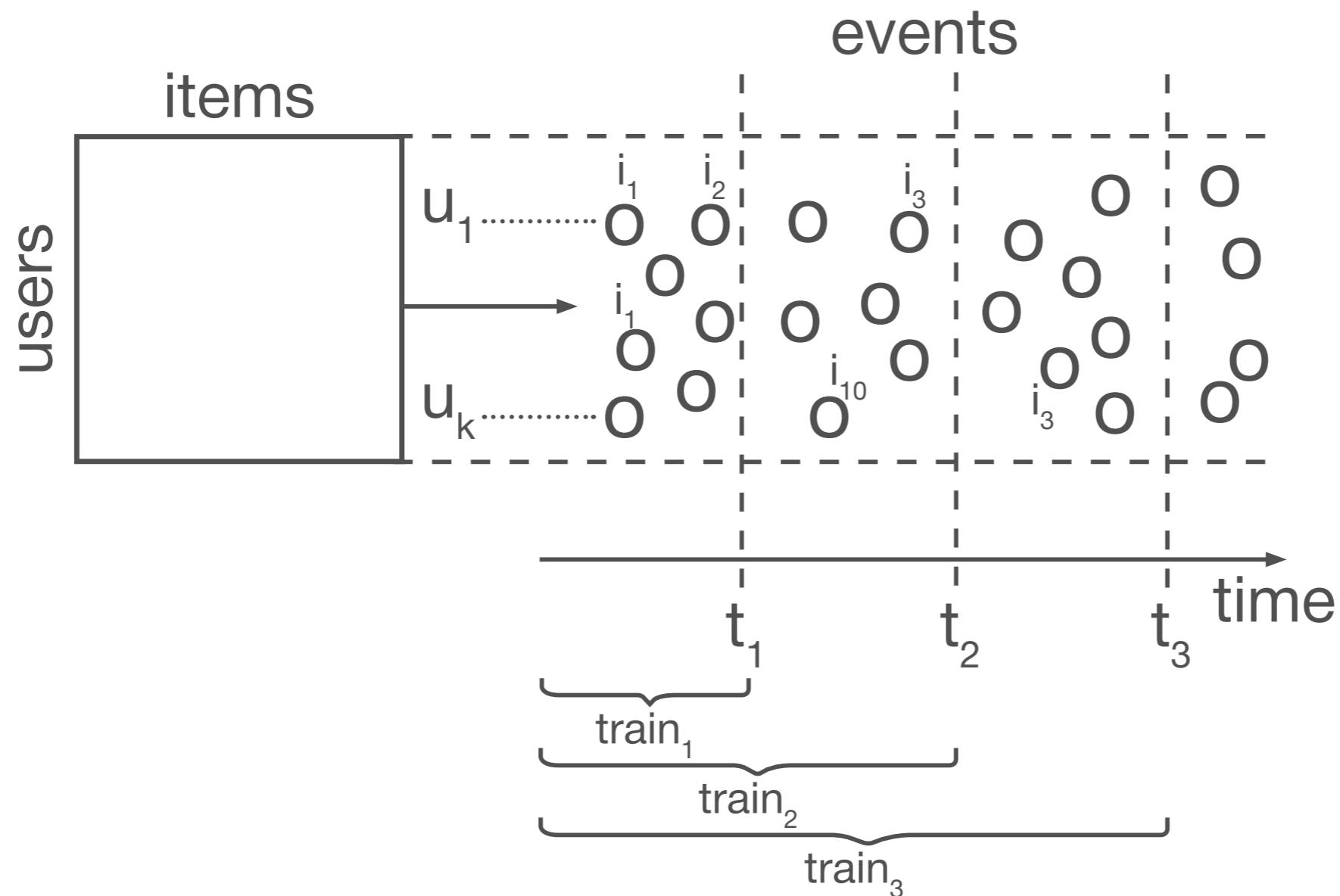
RecSys: Cross Validation



RecSys: Cross Validation



RecSys: Cross Validation



Introduction into RS: Goals

- **explain** basic ideas of Content-Based (**CB**), Collaborative Filtering (**CF**) and Knowledge-Based (**KB**) Recommender System algorithms;
- know steps to **train** simple RS algorithms;
- learn how to **design** RS experiment and **evaluate** it;
- learn how to **overcome** cold-start problem;

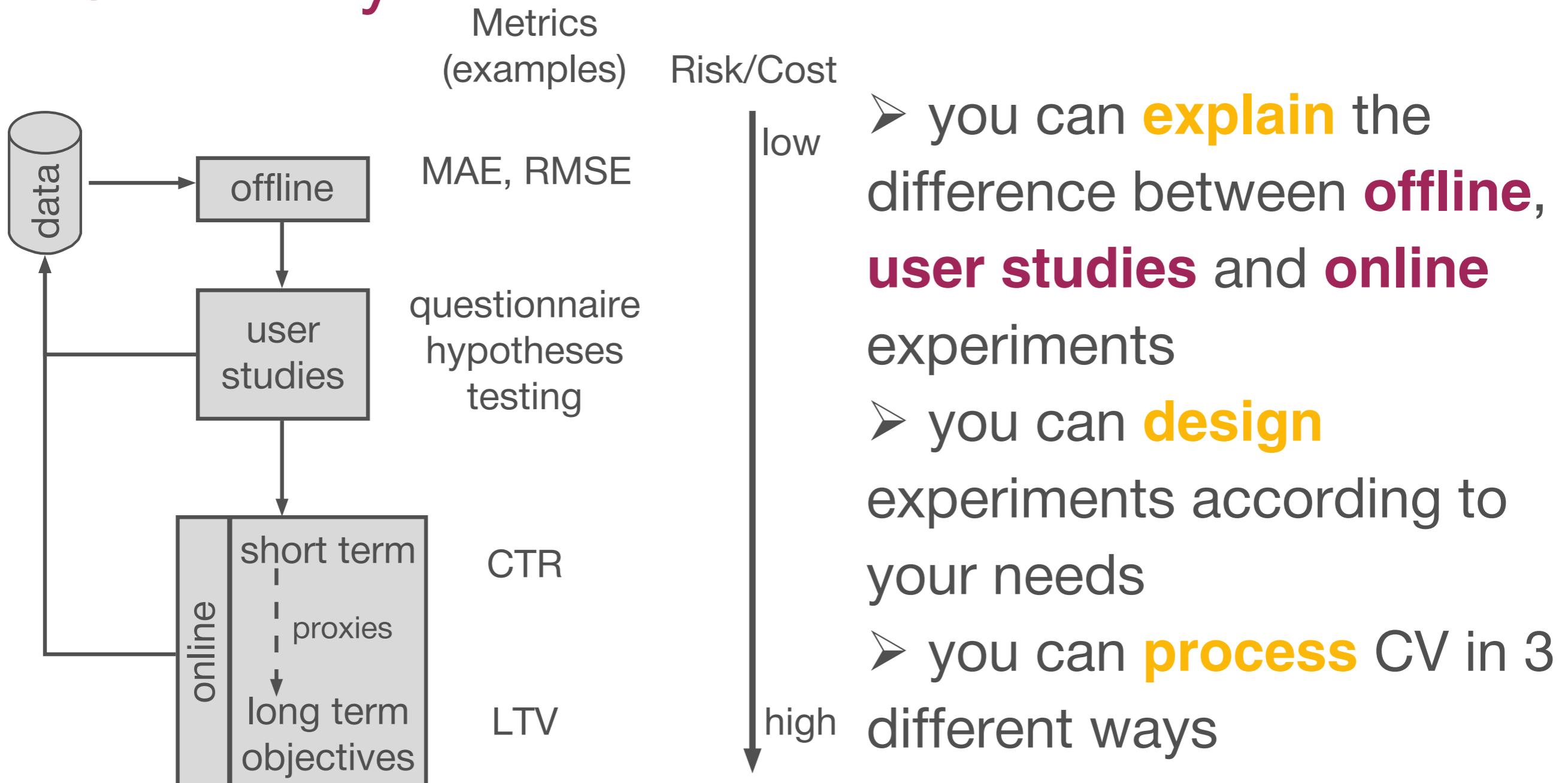
Summary

- you can **build** the following Non-Personalized Recommender System:
 - «Most popular items»
 - «Trending items»
 - «Users who likes A item also likes B item»

Summary

- you can **build** the following Content-Based Recommender System taking into account:
 - «User / Item representation»
 - «Distance / similarity metric»
- you can **explain** pros / cons of Content-Based RS

Summary



References

Теоретический минимум:

1. лекция Introduction to Machine Learning 10-701 CMU 2015 (Alex Smola);
2. лекция Воронцова про коллаборативную фильтрацию;

Учебная программа:

1. лекции по RS на Coursera (последняя неделя в рамках курса "Big Data Applications: Machine Learning at Scale");
2. Xavier Amatriain Lectures 1+2;
3. Xavier Amatriain Lectures 3+4

References (Deep Dive)

1. специализация про RS на Coursera (University of Minnesota)
2. книги:
 - Recommender System Handbook (2015, Ricci, Francesco, Rokach, Lior, Shapira, Bracha (Eds.))
 - Recommender Systems (2016, Aggarwal, Charu C.)
 - Practical Recommender Systems (2015-2017, Kim Falk)
3. Блог <https://buildingrecommenders.wordpress.com/>
4. материалы конференций ACM RecSys

Feedback

лекции: <https://goo.gl/forms/09CxzkyxWrJ2hLlw1>

семинары: <https://goo.gl/forms/SIa0Elv5PrrEnwwa2>

Thank you! Questions?



Appendix

Applications and Evaluation

Evaluation Metrics (formulas)

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

Accuracy

$$DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2(i)}$$

IDCG is the value for the perfect ranking
 rel_i is 1 if the user watched the program at rank i
 or 0 otherwise

$$d(i,j) = \begin{cases} \frac{1}{3} \text{ category}(i=j) \\ \frac{1}{3} \text{ subcategory}(i=j) \\ \frac{1}{3} \text{ channel}(i=j) \end{cases}$$

$$ILD(u) = \frac{1}{|R|(|R|-1)} \sum_{i \in R} \sum_{j \in R} d(i,j) \quad ILD = \frac{1}{|U|} \sum_{u \in U} ILD(u)$$

Novelty

$$MSI(u) = \frac{1}{|R_u|} \sum_{i \in R} \frac{|\{v \in U \mid i \in W_v\}|}{|U|}$$

$$MSI = \frac{1}{|U|} \sum_{u \in U} MSI(u)$$

Serendipity

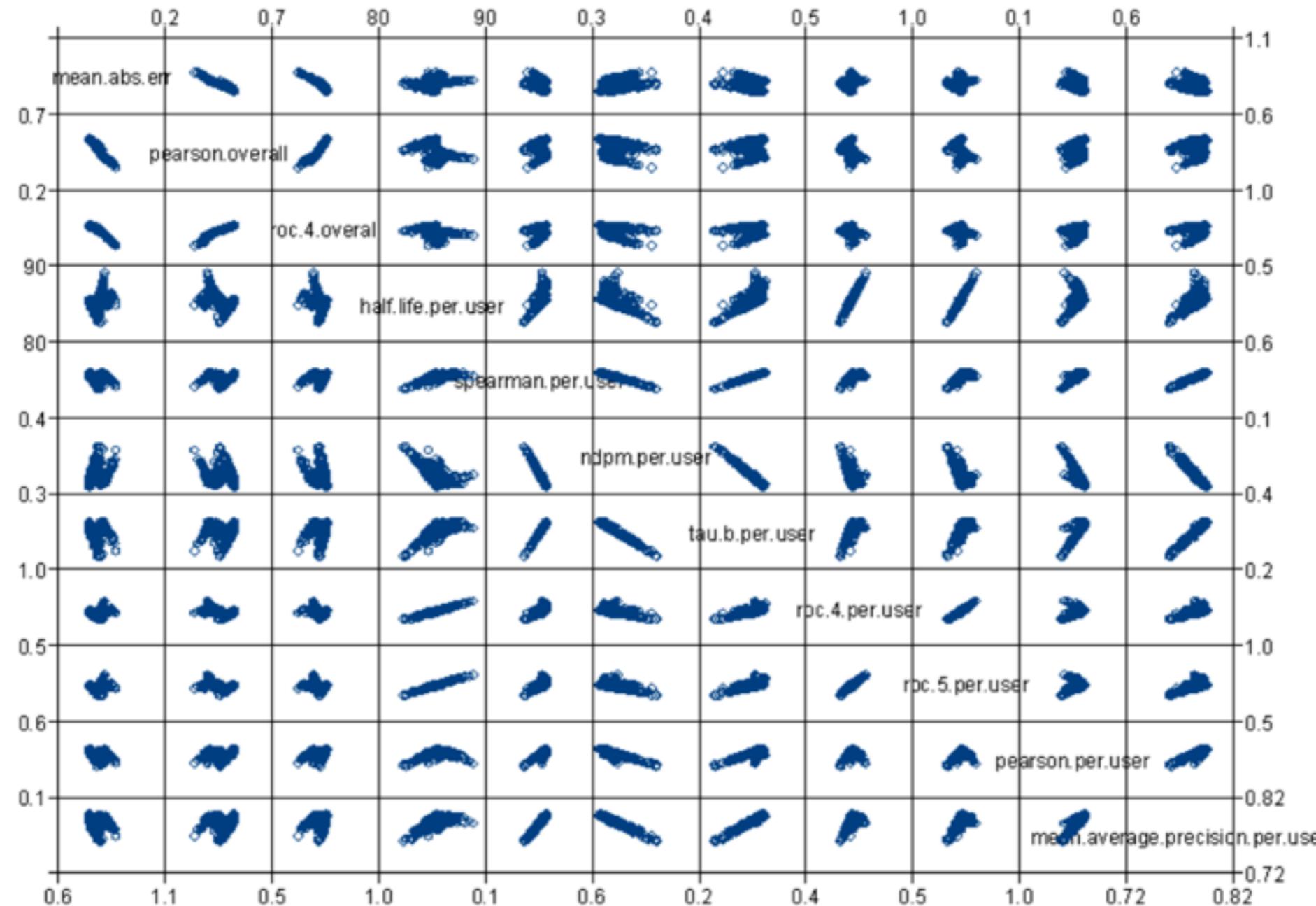
$$Unexp(u) = \frac{1}{|R||W_u|} \sum_{i \in R} \sum_{j \in W_u} d(i,j) \quad Unexp = \frac{1}{|U|} \sum_{u \in U} Unexp(u)$$

R_u = list of recommended items for user u

W_u = watching history for user u

U = list of users

Applications and Evaluation



Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen and John T. Riedl
(2004), Evaluating Collaborative Filtering Recommender Systems, ACM TOIS

Контрольные вопросы

1. Запишите функционал для оптимизации задачи предсказания рейтингов. Какие существуют модификации?
2. Запишите функционал для оптимизации задачи рекомендации ТОП-5 товаров.
3. В каких областях имеет смысл применять Knowledge-Based рекомендательные системы?
4. Запишите формулу для учета "устаревания" рейтингов и приведите пример использования в реальной жизни.
5. Что такое проблема холодного старта? Какие способы борьбы с ним?
6. Запишите 3 формулы для предсказания рейтингов в случае Content-Based рекомендательных систем.