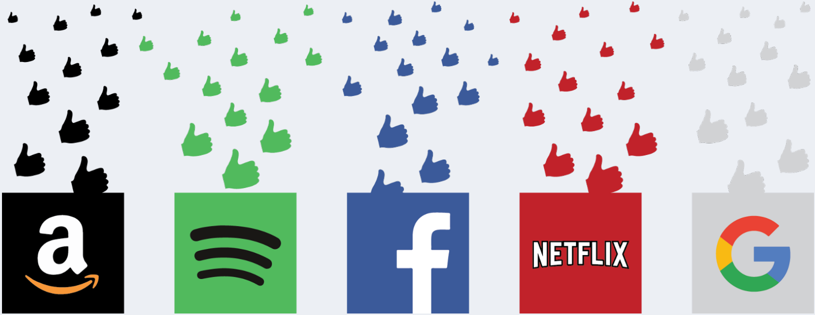
STAT3009 Recommender Systems

Lec1: Overview of Recommender Systems

by Ben Dai (The Chinese University of Hong Kong)
on Department of Statistics

» Big Picture: Recommender Systems

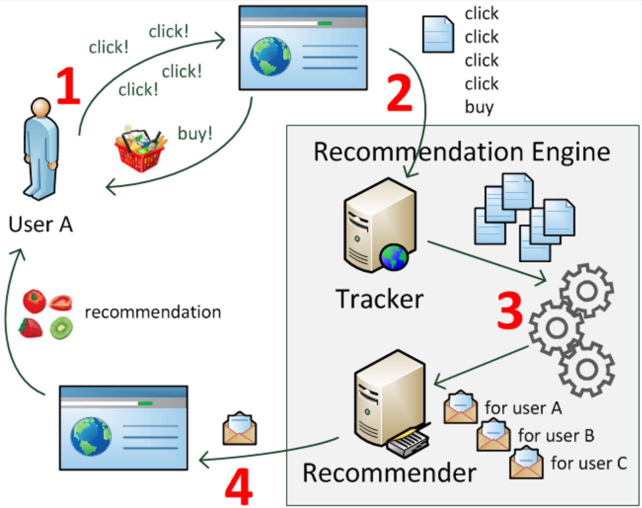
Netflix, YouTube, Taobao and Amazon are all examples of recommender systems (RSs) in use. The systems recommend users with relevant items (suggestions) based on **users' historical data**.



Source[1]

[1] thedatascientist.com/right-way-recommender-system-startup

» Big Picture: Recommender Systems



Source[2]

[2] medium.com/voice-tech-podcast/a-simple-way-to-explain-the-recommendation-engine-in-ai-

» Advantages: Recommender Systems

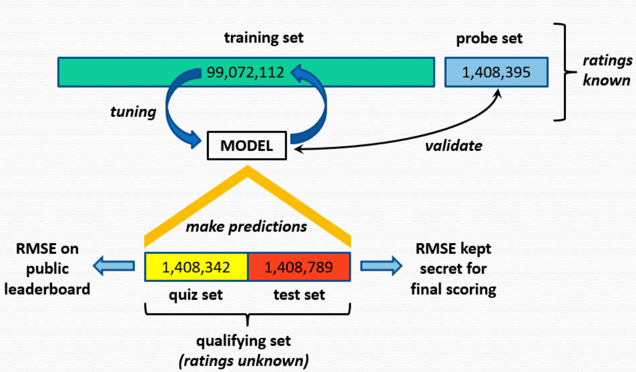
* **Revenue Increase.** The successful recommendation systems lead to the **29% annual sales increase** for Amazon[3].
* **User Satisfaction Increase.** A good RS could offer the personalized suggestions and understand the users' needs.
* **Other Application.** Financial products, medicine recommender systems, ...

That's why the RS is one of the **most valuable** applications of Machine Learning (ML).

[3] https://azati.ai/recommendation-systems-benefits-and-issues/

» Example: Netflix Prize

* **Netflix Prize Competition:** 51051 contestants on 41305 teams from 186 different countries
* $1 million prize for **10% improvement** on Netflix
* **Competition Structure**



Source[4]

[4] https://pantelis.github.io/cs301/docs/common/lectures/recommenders/netflix/

» Dataset: Netflix Prize

* **Netflix Prize Competition**
* **Training dataset**[5]: [**MovieIDs**, **CustomerIDs**, **Ratings**]
    * **MovieIDs** range from 1 to 17770 sequentially.
    * **CustomerIDs** range from 1 to 2649429, with gaps. There are 480189 users.
    * 100 million **Ratings** are on a five star (integral) scale from 1 to 5.
    * **InClass demo: load Netflix dataset via Python**
* **Testing dataset:** [**MovieIDs**, **CustomerIDs**]
* **Evaluation:** The root mean squared error (RMSE)

We need a **formal mathematical model** to formulate a RS problem.

[5] https://www.kaggle.com/netflix-inc/netflix-prize-data

» Data: Recommender Systems

	Gladiator	Space Jam	Pitch Perfect	Life of Pi	Dear Basketball
Rajon	?	?	3	?	?
James	5	5	?	?	5
Davis	?	?	?	4	5
Dwight	?	?	4	?	5
Bryant	?	?	5	4	5

* **Goal:** Can you predict the rating with question mark?

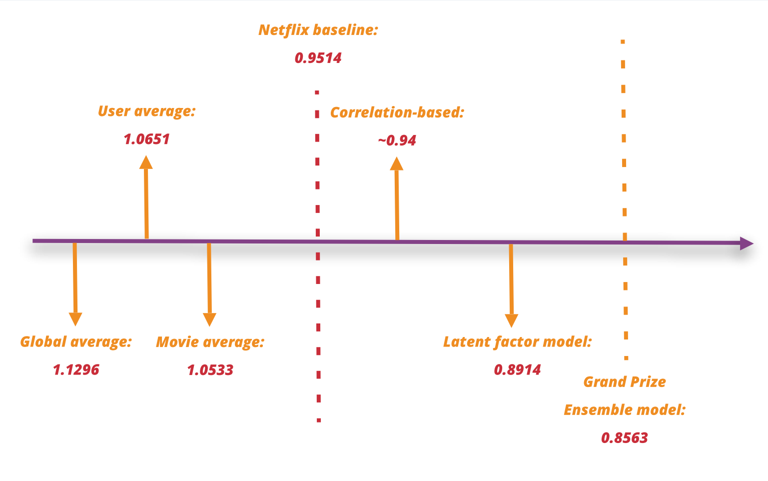
» Formal Model: Recommender Systems

* **User:** $u = 1, \cdots, n$ with n is number of users
* **Item:** $i = 1, \cdots, m$ with m is number of items
* **Rating:** r_{ui} is the u-th user rating in the i-th item.
* **Obs Index set:** $(u,i) \in \Omega$ if the rating for (u,i) pair is observed.
* **InClass demo: find corresponding params in the Python**
* **Evaluation:** Given a testing index set Ω^{te} (set of user-item pairs we want to predict),

$$RMSE = \Big(\frac{1}{|\Omega^{\text{te}}|} \sum_{(u,i)\in\Omega^{\text{te}}} \big(\hat{r}_{ui} - r_{ui} \big)^2 \Big)^{1/2}.$$

* **Goal:** Find predicted ratings $(\hat{r}_{ui})_{(u,i)\in\Omega^{\text{te}}}$ such that **minimizes RMSE**

» Methods: Netflix Prize



* We will cover all methods in this course.

» Baseline methods: Recommender Systems

- **Global average:** predict new ratings by average over all observed ratings

$$\bar{r} = \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} r_{ui}, \quad \hat{r}_{ui} = \bar{r}$$

- **User average:** predict new ratings by average over user's observed ratings

$$\bar{r}_u = \frac{1}{|\mathcal{I}_u|} \sum_{i\in\mathcal{I}_u} r_{ui}, \text{ for } u = 1, \cdots, n; \quad \hat{r}_{ui} = \bar{r}_u,$$

where $\mathcal{I}_u = \{i : (u, i) \in \Omega\}$ is the index set for observed ratings of the u-th user.

» Baseline methods: Recommender Systems

- **Item average:** predict new ratings by average over item's observed ratings

$$\bar{r}_i = \frac{1}{|\mathcal{U}_i|} \sum_{u \in \mathcal{U}_i} r_{ui}, \text{ for } i = 1, \cdots, m; \quad \hat{r}_{ui} = \bar{r}_i,$$

  where $\mathcal{U}_i = \{u : (u, i) \in \Omega\}$ is the index set for observed ratings of the i-th item.
- Space-Time Trade-off
- **InClass demo: Implement the baseline models by Python**

» Baseline methods: Recommender Systems

* User-item average

$$\hat{r}_{ui} = \bar{r} + \mu_u + \mu_i$$

where

$$\mu_u = \frac{1}{|\mathcal{I}_u|} \sum_{i \in \mathcal{I}_u} (r_{ui} - \bar{r}), \quad \mu_i = \frac{1}{|\mathcal{U}_i|} \sum_{u \in \mathcal{U}_i} (r_{ui} - \bar{r} - \mu_u)$$

* Interpretation by multi-stage prediction

» Discussion: baseline methods

"All models are wrong, but some are useful." — George E. P. Box

We need to figure out the **assumptions** for each method!

* **Global average** assumes that all users and items are essentially same
* **User average** assumes that a user has equal preference to all items
* **Item average** assumes that all users like "good" items
* **User-item average** assume that additive effects from users and items, **no interaction**

Which Assumption Is More **Realistic**? Why?

Thank you!