# Clustering and Constructing User Coresets to Accelerate Large-scale Top-$K$ Recommender Systems

**Jyun-Yu Jiang***, Patrick H. Chen*, Cho-Jui Hsieh and Wei Wang
(*Equal Contribution)

University of California, Los Angeles (UCLA)

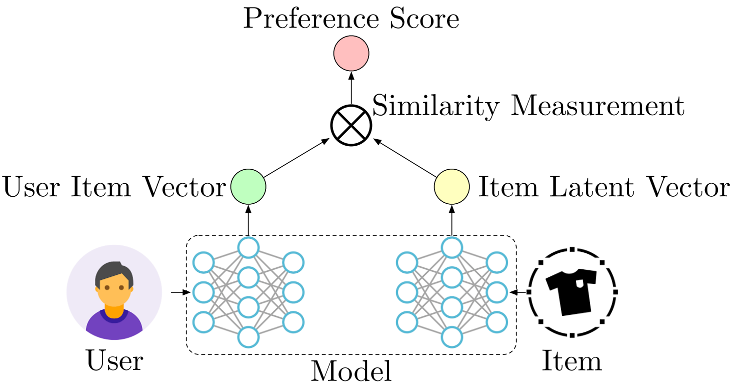
April 22, 2020

# Outline

# Recommender systems are ubiquitous in our lives.

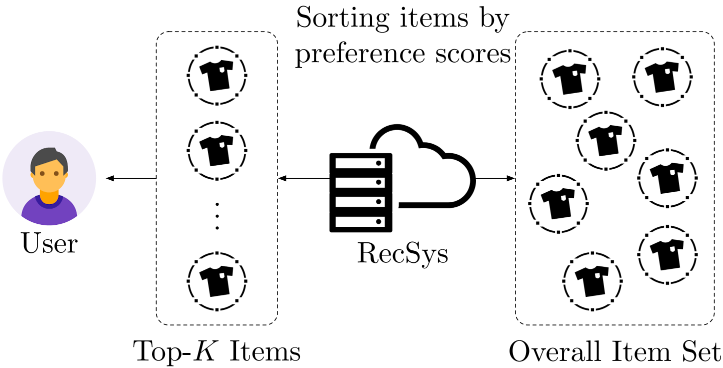
RecSys can suggest users preferred items that can be anything!

# Conventional Approaches for Recommender Systems



# Top-$K$ Recommender Systems

For each user, the systems provide $K$ items with highest preference scores.



# Training Stage v.s. Prediction Stage

## Training Stage

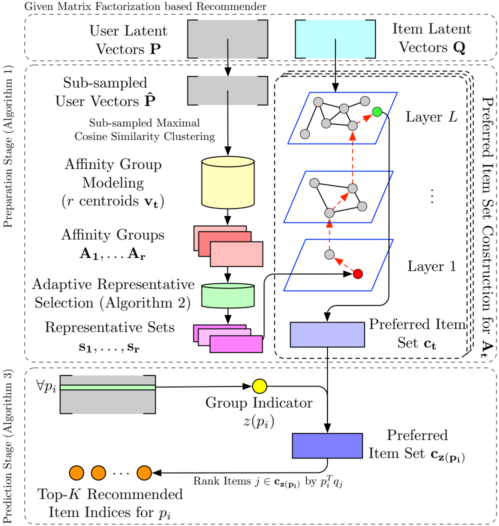
- Training data are limited.
- Negative examples are sampled.
- Fast speed.
- Customers don't care.
- Could be minutes.

## Prediction Stage

- User-item pairs are exhaustive.
- All items should be considered.
- Slow speed.
- Customers care.
- Could be days.

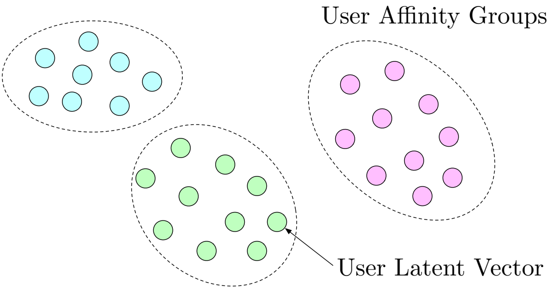
Can we accelerate the prediction stage?

# Framework Overview



# Affinity Group Modeling by User Clustering

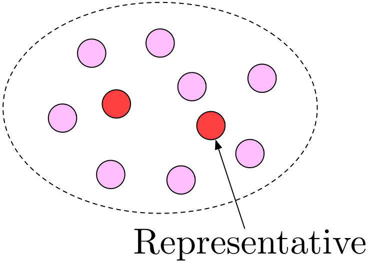
Users in the same affinity group can have similar interests.




However, interests can be still diverse while a group can have many users.

# Coreset Vectors as Representations

A coreset of user vectors may cover the interests of same-group users.




## $\delta$-user Coreset

$$\left| p_i q^T - \mathcal{N}_{\boldsymbol{s}_t}(p_i)\, q^T \right| \leq \delta,$$

where $\mathcal{N}_{\boldsymbol{s}_t}(p_i) \in \boldsymbol{s}_t$ is the nearest coreset representative for $p_i$; $\delta > 0$ is a small enough constant

# Adaptive Representative Selection

**Algorithm 2:** Adaptive Representative Selection

**Input:** User latent vectors for an affinity group $\boldsymbol{P}$, the number of iterations $T$, the threshold $\epsilon$, the number of new representatives $w$ ;

**Output:** Representative vectors $\boldsymbol{s}$.

Initialize $\boldsymbol{s} = \emptyset$ ;

$\mathrm{I} = \arg\max_t \boldsymbol{s}^T \boldsymbol{P}$ ;

**repeat**

    **for** $i = 1 \ldots |\mathbf{s}|$ **do**

        $\boldsymbol{s}_i = \sum_{j \in \{j | I[j] = i\}} \boldsymbol{P}[j]$ ;

        $\boldsymbol{s}_i = \boldsymbol{s}_i \ / \ \|\boldsymbol{s}_i\|_2$ ;

    $\mathrm{I} = \arg\max_t \boldsymbol{s}^T \boldsymbol{P}$ ;

    Outliers = $\{j | \boldsymbol{s}_{I[j]}^T \boldsymbol{P}_j < \epsilon\}$ ;

    **for** $j \in$ *Outliers* **do**

        Draw $i$ from $1 \ldots w$ ;

        $\mathrm{I}[j] = |\mathbf{s}| + \mathrm{i}$ ;

    **if** *Outliers* $\neq \emptyset$ **then**

        Append $w$ vectors to $\boldsymbol{s}$ ;

**until** *Outliers* $= \emptyset$;

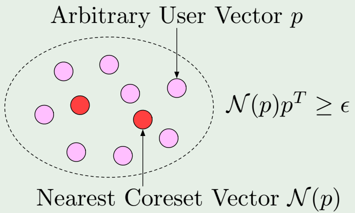
Outliers = $\{j | \boldsymbol{s}_{I[j]}^T \boldsymbol{P}_j < \epsilon\}$ ;

Append $\boldsymbol{P}_{\text{Outliers}}$ to $\boldsymbol{s}$ ;

**return** $\boldsymbol{s}$.

# Coreset Construction as Finding a Set Cover

## $\epsilon$-Set Cover




## Theorem 1

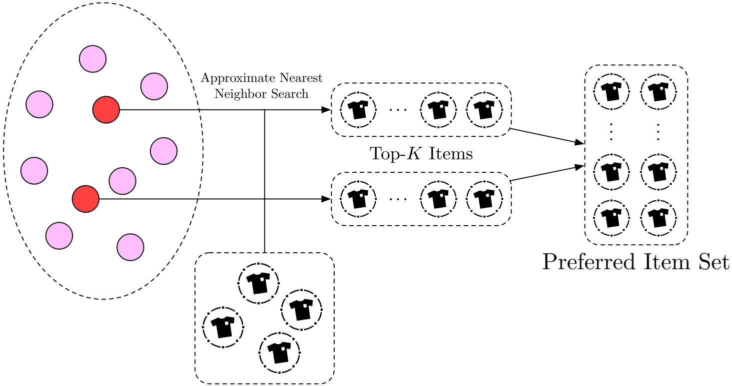
Given an $\epsilon$-cover $\boldsymbol{s}_t$, there exists a $\delta$ such that $\epsilon$-cover $\boldsymbol{s}_t$ is a $\delta$-user coreset of the affinity group.

## Theorem 2

For an affinity group $\boldsymbol{A}_t$, given any item vector $q$, an $\epsilon$-cover of $k$ samples $\{p_i\}$ drawn from $\boldsymbol{P}_{\boldsymbol{A}_t}$ would satisfy following inequality with probability at least $1-\gamma$:

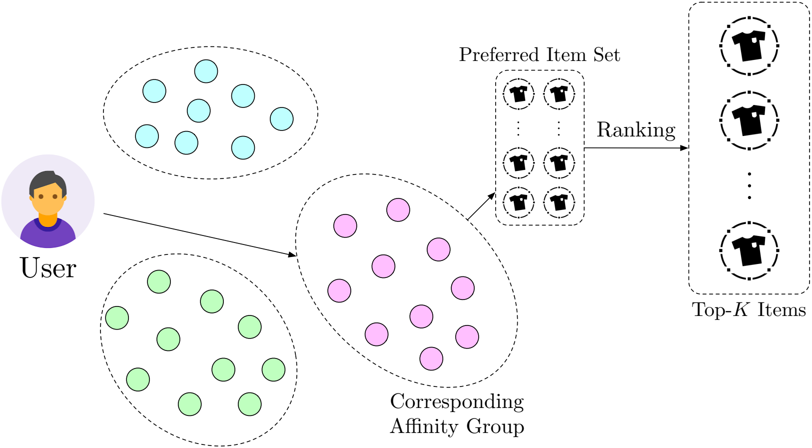
$$\min_i \left( \left| \mathcal{N}(p_i)\, q^T - p_t q^T \right| \right) \leq \delta + \sqrt{\frac{2\log(1/\gamma)}{k}}.$$

# Preferred Item Set Construction




Any approximate nearest neighbor search method is applicable.

# Prediction Stage



# Experimental Datasets

| Task | Item Recommendation | | |
|---|---|---|---|
| Dataset | MovieLens | Last.fm | Amazon |
| #(Users) | 138,493 | 359,293 | 2,146,057 |
| #(Items) | 26,744 | 160,153 | 1,230,915 |

| Task | Personalized Link Prediction | | |
|---|---|---|---|
| Dataset | YouTube | Flickr | Wikipedia |
| #(Users) | 1,503,841 | 1,580,291 | 1,682,759 |
| #(Items) | 1,503,841 | 1,580,291 | 1,682,759 |

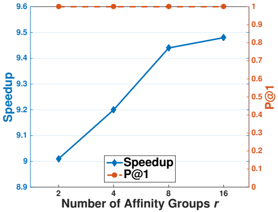
# Experimental Settings

- We train a non-negative matrix factorization model as ground truths.
- Experimental methods aims at providing top-$K$ items for all users.
- Evaluated with
  - Speedup rate (SU) compared to the $O(m\text{users} \times n\text{items})$ approach
  - Split of preparation time (PT) and inference time (IT)
  - Precision at 1 (P@1) and 5 (P@5)
- See more details in our paper.

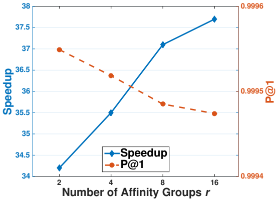
# Top-$K$ Recommendation Results

| Task | Item Recommendation | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | MovieLens | | | | | Last.fm | | | | | Amazon | | | | |
| Method | SU | PT | IT | P@1 | P@5 | SU | PT | IT | P@1 | P@5 | SU | PT | IT | P@1 | P@5 |
| $\epsilon$-Approx | 0.7x | 0.19s | 99.00s | 0.753 | 0.671 | 0.5x | 1.40s | 36.78m | 0.378 | 0.467 | 0.2x | 23.42s | 107.34h | 0.529 | 0.559 |
| GMIPS | 3.9x | N/A | 18.41s | 1.000 | 0.972 | 2.3x | N/A | 7.55m | 0.997 | 0.966 | 1.8x | N/A | 14.57h | 0.993 | 0.952 |
| SVDS | 1.0x | 0.10s | 69.00s | 1.000 | 1.000 | 0.9x | 0.10s | 19.25m | 0.984 | 0.984 | 1.3x | 5.32s | 19.46h | 0.952 | 0.953 |
| FGD | 2.8x | 4.94s | 20.10s | 1.000 | 0.999 | 10.9x | 0.49m | 1.07m | 0.997 | 0.988 | 19.7x | 42.76m | 35.83m | 0.986 | 0.977 |
| L2S | 3.0x | 22.15s | 1.72s | 1.000 | 1.000 | 9.0x | 1.77m | 0.12m | 0.993 | 0.980 | 21.2x | 71.02m | 1.86m | 0.988 | 0.979 |
| CANTOR | **9.4x** | 6.17s | 1.36s | 1.000 | 0.999 | **37.1x** | 0.37m | 0.09m | 0.999 | 0.998 | **29.0x** | 52.13m | 1.26m | 0.994 | 0.991 |
| Task | Personalized Link Prediction | | | | | | | | | | | | | | |
| Dataset | YouTube | | | | | Flickr | | | | | Wikipedia | | | | |
| Method | SU | PT | IT | P@1 | P@5 | SU | PT | IT | P@1 | P@5 | SU | PT | IT | P@1 | P@5 |
| $\epsilon$-Approx | 0.1x | 0.3m | 129.2h | 0.364 | 0.432 | 0.4x | 0.29m | 53.44h | 0.545 | 0.581 | 0.2x | 0.39m | 130.61h | 0.374 | 0.480 |
| GMIPS | 1.4x | N/A | 11.12h | 0.987 | 0.965 | 2.0x | N/A | 10.10h | 0.987 | 0.962 | 3.6x | N/A | 5.64h | 0.991 | 0.974 |
| SVDS | 1.0x | 0.03m | 15.30h | 0.965 | 0.963 | 1.4x | 0.03m | 14.00h | 0.952 | 0.946 | 1.4x | 0.03m | 14.83h | 0.949 | 0.944 |
| FGD | 44.8x | 10.28m | 10.85m | 0.989 | 0.981 | 37.5x | 17.61m | 14.25m | 0.985 | 0.980 | 93.7x | 4.18m | 8.76m | 0.990 | 0.985 |
| L2S | 6.9x | 135.93m | 0.79m | 0.984 | 0.968 | 8.3x | 142.84m | 0.58m | 0.989 | 0.980 | 22.4x | 53.38m | 0.84m | 0.988 | 0.968 |
| CANTOR | **112.7x** | 7.75m | 0.65m | 0.993 | 0.985 | **54.7x** | 21.31m | 0.53m | 0.994 | 0.990 | **355.1x** | 2.45m | 0.97m | 0.995 | 0.991 |

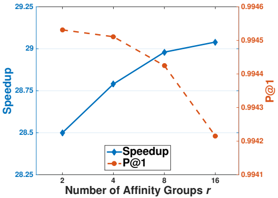
# Number of Affinity Groups



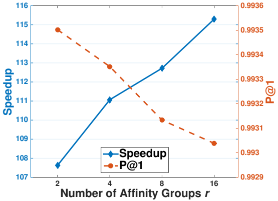

(a) MovieLens



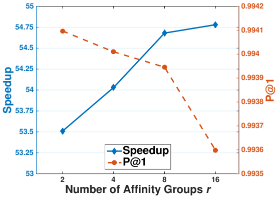

(b) Last.fm



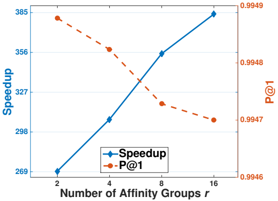

(c) Amazon




(d) YouTube

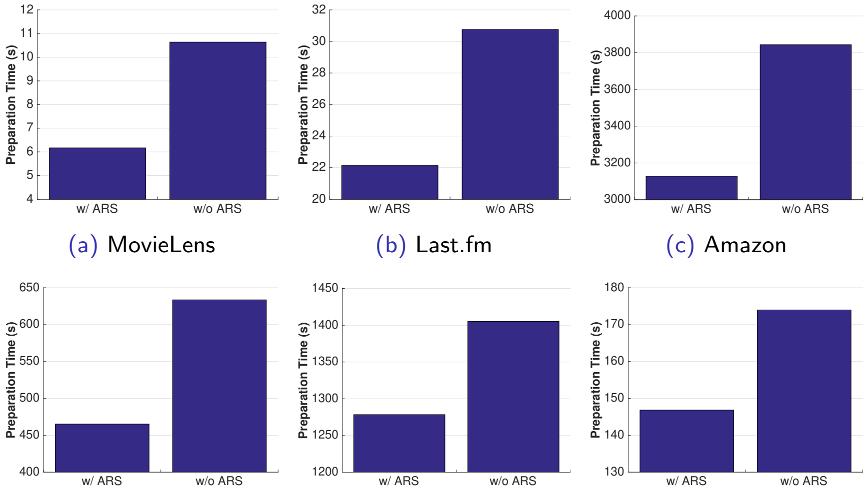



(e) Flickr




(f) Wikipedia

# Effectiveness of Adaptive Representative Selection (ARS)




(a) MovieLens (b) Last.fm (c) Amazon

(d) YouTube (e) Flickr (f) Wikipedia

# Conclusions

- We propose a novel approach to accelerate inference of top-$K$ recsys.
- User affinity groups and representatives save lots of computations.
- Representative coresets as a set cover are theoritically guaranteed.
- Significant improvements on extensive experiments with 6 datasets.
- Analysis shows the effectiveness and robustness of our approach.

Questions? Or ask me by email: jyunyu@cs.ucla.edu