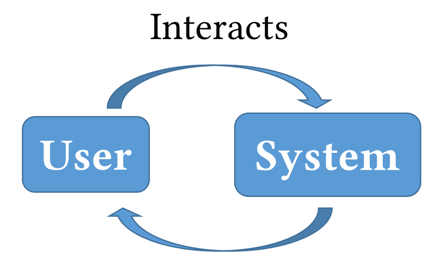
A Simple Multi-Armed Nearest-Neighbor Bandit for Interactive Recommendation

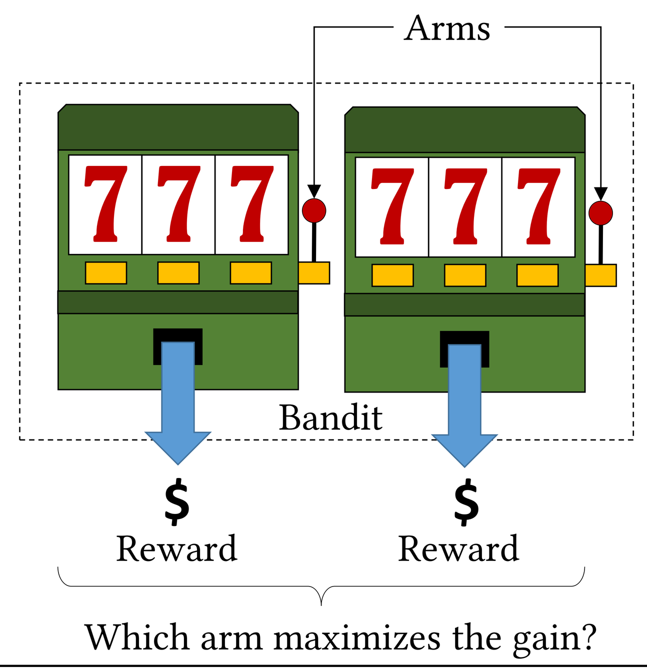
Universidad Autónoma de Madrid

Javier Sanz-Cruzado, Pablo Castells and Esther López
Universidad Autónoma de Madrid
{javier.sanz-cruzado,pablo.castells}@uam.es
esther.lopezramos@estudiante.uam.es

IRG
IR Group @UAM

Motivation





Interactive recommendation

- More realistic perspective
- Recommendation operates in cycles
  - Users react to and interact with recommendations
  - The system gains further input from this interaction
  - The input is used in the following recommendations

Multi-armed bandits

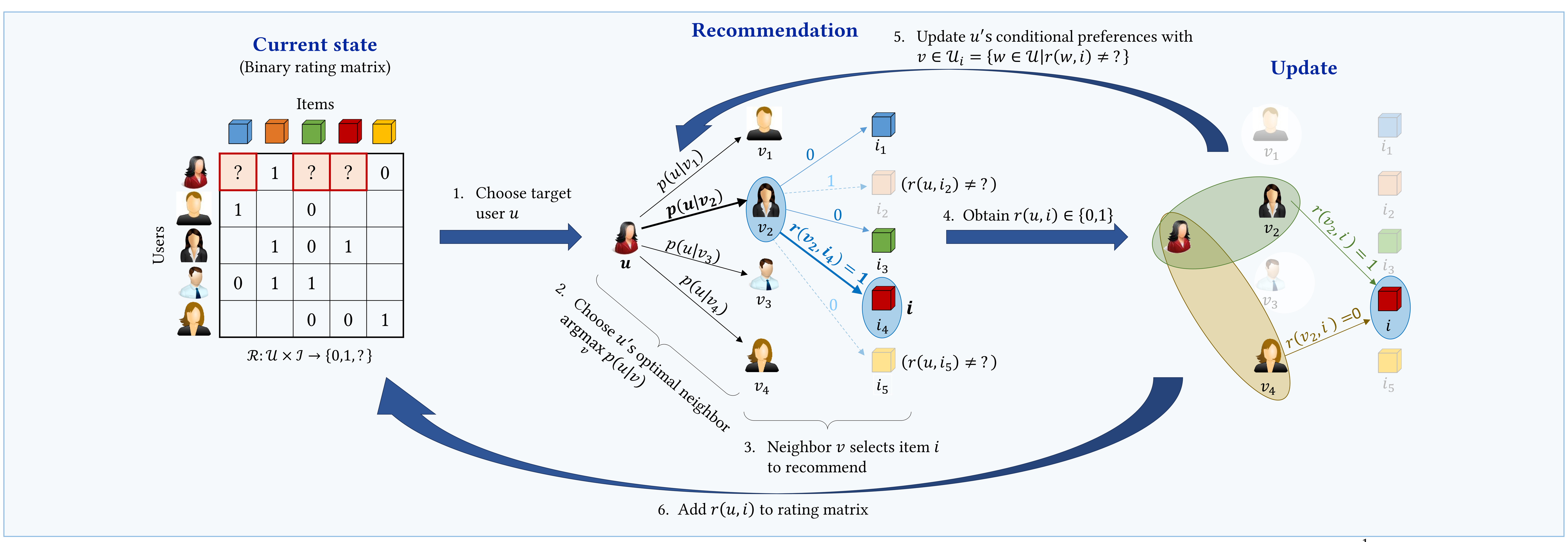
- Select the best among several actions (arms)
- Exploration vs. exploration
  - Select arm with highest estimated value (**exploit**)
  - Select arm to gain knowledge (**explore**)
- Algorithms: ε-greedy, UCB1, Thompson Sampling, etc.

Bandit Recommender Systems

- Use of bandit algorithms to generate recommendations
- Usually arms = items
- **Personalized approaches:** contextual bandits
  - Stochastic versions of Probabilistic Matrix Factorization (PMF)
  - Clusters of users/items
- **Our approach**: user-based kNN with stochastic (bandit-based) neighborhood selection

Recommender bandits	Multi-armed bandits	Our approach
Items $i \in \mathcal{I}$	**Arms**	Neighbors $v \in \mathcal{U}$
Ratings $r(u,i)$	**Rewards**	Ratings $r(u,i)$
Metric (e.g. CTR)	**Estimated arm value**	Cond. preference $P(u\|v)$
Target user $u \in \mathcal{U}$	**Context**	Target user $u \in \mathcal{U}$

Nearest-neighbor bandit

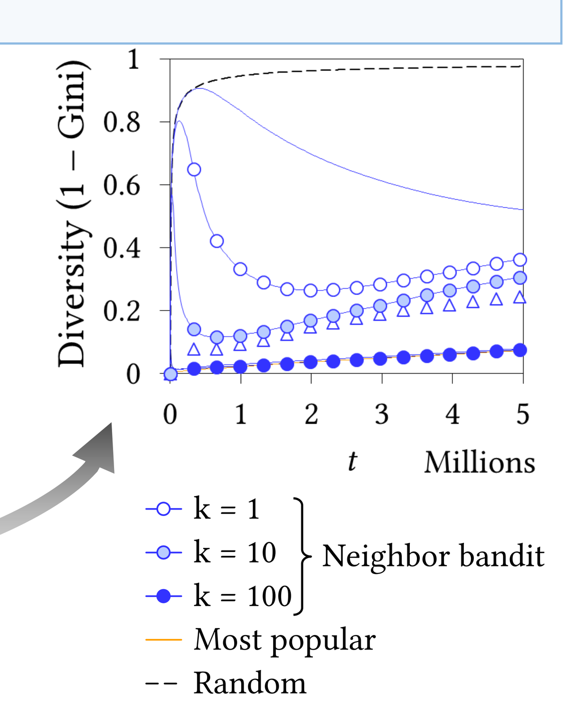


Neighbor bandit

- Given a user u, choose the best neighbor v
- **Conditional preference:** Instead of similarity, $P(u|v)$ probability that u likes an item that v likes
- **Thompson sampling bandit**
  - Rewards considered as a Bernoulli distribution with mean $P(u|v)$
  - $P(u|v)$ estimated from data using posterior Beta distributions, $p(u|v) \sim B(\alpha(v|u), \beta(v|u))$
  - **Hits $\alpha(v|u)$:** Positive common ratings between u and v: $\alpha(v|u) = \alpha_0 + \sum_{i \in \mathcal{I}} r(u,i) r(v,i)$
  - **Misses $\beta(v|u)$:** v's positive ratings not shared by u. $\beta(v|u) = \beta_0 + \sum_{i \in \mathcal{I}} r(v,i) - \alpha(v|u)$
  - Select v maximizing $p(u|v)$
- Recommend the item i that v likes most (ties randomly broken)
- Update $\alpha(w|u)$ for all users $w \in \mathcal{U}$ who have rated the recommended item i

Extension: k neighbors

- Given a target user u, select the best k neighbors
- Multiple play Thompson sampling bandit
- Pick $\mathcal{N}_k(u)$: the k users maximizing $p(u|v)$
- Recommend the item maximizing: $\arg\max_{i \in \mathcal{I}} \sum_{v \in \mathcal{N}_k(u)} p(u|v) r_v(i)$
- Again, update $\alpha(w|u)$ for all users $w \in \mathcal{U}$ who have rated i
- Penalizes exploration in favor of exploitation



Experiments

Data

- **Implicit feedback datasets**
  - 2 Twitter follows graphs (1 month, 200 tweets)
  - 2 FourSquare venue recommendation datasets (New York, Tokyo)
- **Explicit ratings dataset:** MovieLens 1M
  - We take ratings ≥ 4 as positive ($r(u,i) = 1$)

Dataset	#Users	#Items	#Ratings
FourSquare New York	1,083	38,333	227,428
FourSquare Tokyo	2,293	61,858	573,703
Twitter 1 month	9,511	9,511	650,937
Twitter 200 tweets	9,253	9,253	475,608
MovieLens 1M	6,040	3,706	1,000,209

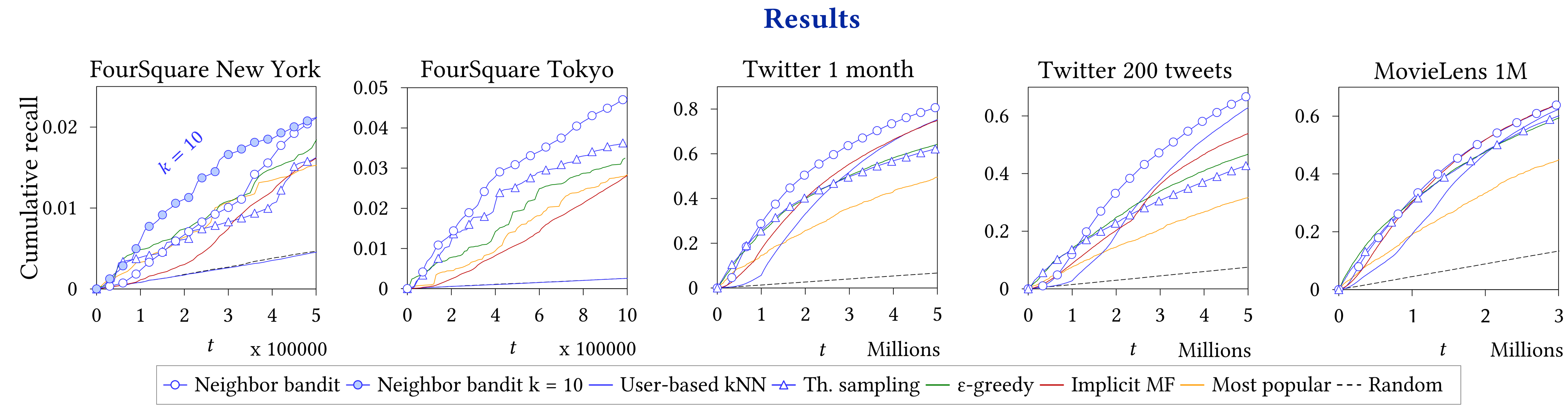
Evaluation approach

- **Offline evaluation:** simulate user feedback using offline data
- **Extreme cold start:** start with no ratings
- Random user selection (one at a time)
- **Metric**: cumulative recall – fraction of discovered positive ratings (growth curve)

Algorithms

- **Bandits**
  - **Our approach:** nearest-neighbors bandit
  - **Item-oriented:** ε-greedy, Thompson sampling
- **Exploitation-only**
  - **Collaborative filtering:** matrix factorization, user-based kNN (cosine)
  - **Sanity-check:** most-popular, random

Results



Full text **Code**

Code for this paper is available at https://github.com/ir-uam/kNNBandit