Bandits Under the Influence

Silviu Maniu, Stratis Ioannidis, Bogdan Cautis

Université Paris-Saclay & Northeastern University

Motivation

Recommender systems: recommending items to users

- preferences may be unknown or highly dynamic
- online recommendations systems – re-learn preferences on the go
- users can be influence by other users – social influence

Objective: online recommendation systems taking into account social influence

- solution framework: sequential learning, multi-armed bandits

Setting – Recommendation

Set of users $[n]$, receiving suggestions at time steps $t \in \mathbb{N}$, each having user profiles $\mathbf{u}_i(t) \in \mathbb{R}^d$

Recommended item: d-dimensional vector $\mathbf{v} \in \mathbb{R}^d$, $\mathcal{B}$ the catalog of recommendable items

Each time step t: user is presented an item i, and presents a rating $r_i(t)$:

$$r_i(t) = \langle \mathbf{u}_i(t), \mathbf{v}_i(t) \rangle + \epsilon$$

Setting – User Preference Evolution

Users are in a social network, and interests evolve in time steps:

$$\mathbf{u}_i(t) = \alpha \mathbf{u}_i^0 + (1-\alpha) \sum_{j \in [n]} P_{i,j} \mathbf{u}_j(t-1), \; i \in [n]$$

- social parameter $\alpha \in [0,1]$
- influence network between users i and j, P_{ij}

Our Contributions

1. Establish the link between the **online recommendation** and **linear bandits**
2. Apply the **non-stationary** setting to the classic LinREL and Thompson Sampling algorithms from the bandit literature
3. Study **tractable cases** for solving the optimizations in each step of the algorithms

Link with Bandits

Want to minimize the aggregate regret:

$$R(T) = \sum_{t=1}^{T} \sum_{i=1}^{n} \langle \mathbf{u}_i(t), \mathbf{v}_i^*(t) \rangle - \langle \mathbf{u}_i(t), \mathbf{v}_i(t) \rangle$$

Bandit setting: we notice that the aggregate reward is a linear function of the matrix of user profiles U^{o}:

- expected reward $\bar{r}(t) = \mathtt{u}_{\mathrm{o}}^{\top} L(t) \mathtt{v}$ – function of vectorized forms of the user and item matrices $\mathtt{u}$, $\mathtt{v}$ and a matrix capturing the social evolution $L(t)$

LinREL – Adapting to Recommendations

LinREL:

- arms are selected from a vector space, and the expected reward observes an linear function of the arm
- to select an armwe use Upper Confidence Bound (UCB) principle – a confidence bound on an estimator
- the unknown model is estimated via least square fit, either L_1 or L_2 ellipsoids

LinREL – Adapting to Recommendations

In our case:

- arms are the items v, modified by $L(t)$ – non-stationary setting
- the estimator is least-squares

$$\hat{\mathrm{u}}_0(t) = \arg\min_{\mathrm{u}\in\mathbb{R}^{nd}} \sum_{\tau=1}^{t-1} \|X(V(\tau), A(\tau))\mathrm{u} - \mathbf{r}(\tau)\|_2^2$$

- recommendations are selected as solution to the non-convex optimization

$$\mathrm{v}(t) = \arg\max_{\mathrm{v}\in\mathcal{B}^{(n)}} \max_{\mathrm{u}\in\mathcal{C}_t} \mathrm{u}^\top L(t)\mathrm{v}$$

- we study the case of $\mathcal{C}^1$, $\mathcal{C}^2$ – ellipsoids in L_1 and L_2

LinREL – Regret

Theorem

Assume that, for any $0 < \delta < 1$*:*

$$\beta_t = \max\left\{128nd \ln t \ln \frac{t^2}{\delta}, \left(\frac{8}{3} \ln \frac{t^2}{\delta}\right)^2\right\}, \tag{1}$$

then, for $\mathcal{C}_t = \mathcal{C}_t^2$*:*

$$\Pr\left(\forall T, R(T) \leqslant n\sqrt{8nd\beta_T T \ln\left(1 + \frac{n}{d}T\right)}\right) \geqslant 1 - \delta, \tag{2}$$

and, for $\mathcal{C}_t = \mathcal{C}_t^1$*:*

$$\Pr\left(\forall T, R(T) \leqslant n^2 d\sqrt{8\beta_T T \ln\left(1 + \frac{n}{d}T\right)}\right) \geqslant 1 - \delta. \tag{3}$$

LinREL – Computational Issues

For $\mathcal{C}^1$ the optimization can be solved **efficiently** for two classes of catalogs:

- **if $\mathcal{B}$ is a convex set** – **convex optimization problem**, need to solve $2n^2d$ convex problems
- **if $\mathcal{B}$ is a finite subset** – can check all $|\mathcal{B}|$ items for a total of $2n^2d$ evaluations

Other Algorithms

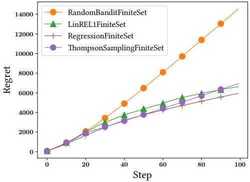
Thompson Sampling

- Bayesian interpretation, assumes a prior on $\mathbf{u}_0$
- in each step, samples this vector from the posterior obtained after the feedback has been observed
- computationally efficient
- **Bayesian regret** of the same order as for LinREL

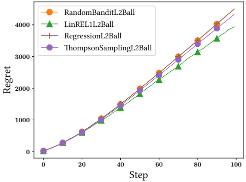
LinUCB

- similar to LinREL, but does not optimize over an ellipsoid
- **non-convex optimization**, inefficient

Results on Synthetic Datasets



(a) Regret, finite set $n = 100$, $d = 20$, $|\mathcal{B}| = 1000$



(b) Regret, L_2 ball $n = 100$, $d = 20$

Synthetic dataset: randomly generated social network, user profiles, and catalog

Results on Real Dataset

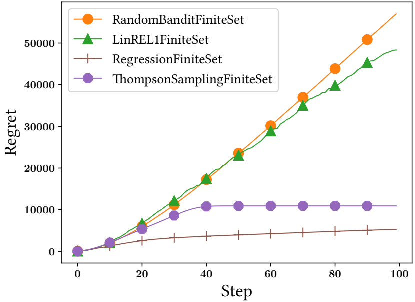


Figure 1: Flixstr regret $n = 206$, $d = 28$, $|\mathcal{B}| = 100$

Flixstr: filtered dataset

- 1 049 492 users in a social network of 7 058 819 links
- 74 240 movies and 8 196 077 reviews