Kernalized Collaborative Contextual Bandits

Leonardo Cella
leonardo.cella@mail.polimi.it
Politecnico di Milano
Milan, Italy

Romaric Gaudel
romaric.gaudel@ensai.fr
Univ. Rennes, Ensai, CREST
Rennes, France

Paolo Cremonesi
paolo.cremonesi@polimi.it
Politecnico di Milano
Milan, Italy

ABSTRACT

We tackle the problem of recommending products in the online recommendation scenario, which occurs many times in real applications. The most famous and explored instances are news recommendations and advertisements. In this work we propose an extension to the state of the art Bandit models to not only take care of different users' interactions, but also to go beyond the linearity assumption of the expected reward. As applicative case we may consider situations in which the number of actions (products) is too big to sample all of them even once, and at the same time we have several changing users to serve content to.

KEYWORDS

Recommender Systems, Contextual Bandits, Kernel Methods

ACM Reference format:
Leonardo Cella, Romaric Gaudel, and Paolo Cremonesi. 2017. Kernalized Collaborative Contextual Bandits. In *Proceedings of RecSys 2017 Posters, Como, Italy, August 27-31,* 2 pages.

1 INTRODUCTION AND RELATED WORKS

In the Recommender Systems (RS) field the most valuable information to rely on are user interactions. That is the reason why Collaborative Filtering methods are the current state of the art model, or at least the ones that give the most important contribution when recommending. In the web we have many real applications such as: computational advertisement, news recommendation or on-line streaming, that do not fit the classical recommending scenario. Their peculiarity is the fact that both the sets of active users and available products are very *fluid*, therefore they change with time [2].

In this on-line recommendation scenario, in particular when we also have *contexts* besides item feature vectors, *multi-armed bandits* techniques have shown to be an excellent solution and are the current state of the art model. Most of the previous efforts on contextual bandits were spent on looking to the recommendation problem from a single user standpoint. We may find just a few preliminary works along the collaborative direction [6].

With this project we also want to consider scenarios where the set of products is too big to be explored entirely, therefore we decide to exploit kernel methods [7]. They provide a way to extract from the primal context features space non-linear relationships that map original features to the obtained rewards relying on similarity information between contexts. It is useful also to mention that there are settings where contexts similarities are the only available information [3]. Previous approaches to the contextual bandit problems usually assume that the features-rewards mapping is a linear relationship ([1], [5], [4]) the only exception is given by [8].

In a nutshell, in this paper we demonstrate that Collaborative Bandits [6] may be extended, through kernel trick, and get out of the linearity assumption. Our modeling assumptions are that the expected reward obtained after choosing a product to recommend can be expressed as a function of both the product features and other users' interactions on different items. To do so, in this project we introduce a contextual multi-armed bandits model that rely on kernel methods to go beyond the linearity assumption, and on graphs to take into account the Collaborative aspect.

The following section details the modeling assumptions and Section 3 presents the proposed algorithm.

2 LEARNING MODEL

We assume that the learning process can be divided in T discrete rounds $t = 1, \dots, T$. At time t, the learner receives a user index $u_t \in \boldsymbol{U} = \{1, \dots, n\}$ to provide recommendation to, and a set of available contexts[1] (arms) $\mathcal{X}_t = \{\mathbf{x}_{t,1}, \dots, \mathbf{x}_{t,c_t}\} \subseteq \mathfrak{R}^d$. The learner has to select one of the content feature vectors $\overline{\mathbf{x}}_t \in \mathcal{X}_t$ to recommend to u_t, and then it observes a payoff v_t whose expectation is $\phi(\overline{\mathbf{x}}_t)^\top \theta_{\mathbf{u_t}}$, where (i) we assume there exists a mapping $\phi : \mathfrak{R}^d \to \boldsymbol{H}$ that maps the data to a Hilbert Space, and (ii) $\theta_{u_t} \in \boldsymbol{H}$ is unknown from the learner. More specifically, we call $\mathfrak{R}^d$ the *primal space* and $\boldsymbol{H}$ the related *reproducing kernel Hilbert space.*

The learner aims at maximizing the total payoff over the T rounds: $\sum_t v_t$. This goal is usually translated in a minimization/bounding problem of a loss variable called *(pseudo-)regret*, that measures the gap of the learner policy wrt. the optimal one (being aware of parameters $(\theta_u)_{u \in \boldsymbol{U}}$). The regret at time t is defined as:

$$r_t = \max_{\mathbf{x} \in \mathcal{X}_t} \phi(\mathbf{x})^\top \theta_{u_t} - \phi(\overline{\mathbf{x}}_t)^\top \theta_{u_t} \tag{1}$$

Let define the *kernel function* as: $k(\mathbf{x}, \mathbf{x'}) := \phi(\mathbf{x})^\top \phi(\mathbf{x'})\ \forall \mathbf{x}, \mathbf{x'} \in \mathfrak{R}^d$. From that function and given a dataset composed by t records $\mathbf{x}_1, \dots, \mathbf{x}_t \in \mathfrak{R}^d$, we define the *kernel matrix* as $K_t := k(\mathbf{x}_i, \mathbf{x}_j)_{i,j \le t}$.

It's worth noting that in such scenario there is no need to get access to content representation. As we clarify in next section, the algorithm only requires to know the kernel value $k(\mathbf{x}, \mathbf{x'})$ for any pair $(\mathbf{x}, \mathbf{x'})$ of contents which have been recommended. Similarly, the estimates of the unknown parameters $(\theta_u)_{u \in U}$ are never explicitly expressed.

In order to represent the collaborative effect, we also assume that users and contents can be co-clustered as expressed in the following. First, for each content vector $\mathbf{x} \in \mathfrak{R}^d$, the set $\boldsymbol{U}$ can be

[1] Along the paper, we identify contexts as the concatenation of item feature representation and real contextual properties(when available). Therefore they fully characterize the available items properties.

RecSys 2017 Poster Proceedings, August 27-31, Como, Italy

clusterized as $C(\mathbf{x}) = \left(\boldsymbol{U}_i^{\mathbf{x}}\right)_{1 \le i \le m(\mathbf{x})}$, such that (i) $\boldsymbol{U} = \bigcup_{i=1}^{m(x)} \boldsymbol{U}_i^{\mathbf{x}}$ and $\boldsymbol{U}_i \cap \boldsymbol{U}_j = \varnothing$ for any $1 \le i < j \le m(\mathbf{x})$, and (ii) the users belonging to the same cluster react similarly when the content with feature vector $\mathbf{x}$ is recommended to them. Namely, if two users u and u' belong to the same cluster $\boldsymbol{U}_k^{\mathbf{x}}$, then $\left|\phi(\mathbf{x})^\top \theta_{\mathbf{u}} - \phi(\mathbf{x})^\top \theta_{\mathbf{u}'}\right| \le \gamma$ for some unknown gap parameter $\gamma \ge 0$.

Second, the content-vectors are themselves clustered in sets $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m$ such that two contents belonging to the same cluster induce the same clustering on $\boldsymbol{U}$: $\forall 1 \le j \le m, \forall \mathbf{x}, \mathbf{x}' \in \boldsymbol{X}_j, C(\mathbf{x}) = C(\mathbf{x}')$.

Clearly the co-clustering mapping is not known and is one of the two main learning objective of the proposed algorithm. The novelty compared to [6] is that we assume clusterings over users is determined by non-linear functions of item features thanks to the applications of kernel methods.

3 ALGORITHM

The proposed algorithm (Algorithm 1) adopts the upper confidence bound paradigm to manage the exploration-exploitation tradeoffs. In detail the estimation of the expected reward and of its corresponding confidence bound is done given the estimations of the clustering, which will be depicted later on. At time-step t, for a user u and an item $\mathbf{x}$, we first define the estimation $\hat{\theta}_{u,t,\mathbf{x}}$ as the solution of the optimization problem

$$\arg\min_{\theta} \sum_{s \in \mathbf{T}_{u,t,\mathbf{x}}} (v_t - \phi(\bar{\mathbf{x}}_t)^\top \theta)^2 + \lambda \|\theta\|^2, \tag{2}$$

where $\mathbf{T}_{u,t,\mathbf{x}}$ is composed of past time-steps s at which the user u_s belongs to the same (estimated) cluster as u (given $\mathbf{x}$). By denoting $\boldsymbol{\Phi}_{u,t,\mathbf{x}}$ the matrix which rows correspond to the vectors $\{\phi(\mathbf{x}_t)^\top, t \in \mathbf{T}_{u,t,\mathbf{x}}\}$, $\mathbf{K}_{u,t,\mathbf{x}}$ the product $\boldsymbol{\Phi}_{u,t,\mathbf{x}} \boldsymbol{\Phi}_{u,t,\mathbf{x}}^\top$, and $\mathbf{r}_{u,t,\mathbf{x}}$ the vector $[r_t, t \in \mathbf{T}_{u,t,\mathbf{x}}]^\top$, we get $\hat{\theta}_{u,t,\mathbf{x}} = \boldsymbol{\Phi}_{u,t,\mathbf{x}}{}^\top (\mathbf{K}_{u,t,\mathbf{x}} + \gamma I)^{-1} \mathbf{r}_{u,t,\mathbf{x}}$. This lead to the following estimate for the expected reward of content $\mathbf{x}$ wrt. user u at time t:

$$\hat{v}_{u,t,\mathbf{x}} = \phi(\mathbf{x})^\top \hat{\theta}_{u,t,\mathbf{x}} = k_{u,t,\mathbf{x}}{}^\top (\mathbf{K}_{u,t,\mathbf{x}} + \gamma I)^{-1} \mathbf{r}_{u,t,\mathbf{x}}, \tag{3}$$

where $k_{u,t,\mathbf{x}} = \boldsymbol{\Phi}_{u,t,\mathbf{x}} \phi(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_t), t \in \mathbf{T}_{u,t,\mathbf{x}}]^\top$. Note that this equation expresses $\hat{v}_{u,t,\mathbf{x}}$ only after past rewards and kernel distance between contents.

Similarly, the confidence interval on-top of $\hat{v}_{u,t,\mathbf{x}}$ is expressed as

$$\hat{\sigma}_{u,t,\mathbf{x}} = \lambda^{\frac{1}{2}} \sqrt{(k(\mathbf{x}, \mathbf{x}) - k_{u,t,\mathbf{x}}^\top (K_t + \gamma I)^{-1} k_{u,t,\mathbf{x}}] \log(t+1)}. \tag{4}$$

Finally, the chosen arm is the one that maximizes the upper confidence bound :

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{X}_t} \hat{v}_{u_t,t,\mathbf{x}} + \eta \hat{\sigma}_{u,t,\mathbf{x}}, \tag{5}$$

where $\eta \ge 0$ is the exploration parameter.

It remains to explain the way the clusterings are estimated. The clusterings are represented by maintaining undirected graphs for which each connected component represents a cluster. One graph stands for contents, and for each contents-cluster induced thereof there is one graph to cluster users. The algorithm starts with fully connected graphs : every contents and every users are reachable each other. Thereafter, after getting feedback v_t, we first delete

Algorithm 1 Collaborative Kernalized Bandits

```
Initialize the user graph as connected over U and the item graph as connected over I
for t = 1,2,... do
    receive u_t ∈ U and the set of contents X_t
    for x ∈ X_t do //Collaborative part
        identify the current user cluster
        compute cluster-aggregated variables v̂_{u,t,x} and σ̂_{u,t,x}
    select recommended content x̄_t according to Equation (5)
    receive payoff v_t
    update clustering graphs
```

edges from the users-graph associated to the selected content. The deleted edges (u_t, u) are whose such that:

$$|\hat{v}'_{u_t,t,\bar{\mathbf{x}}_t} - \hat{v}'_{u,t,\bar{\mathbf{x}}_t}| \ge \eta' \hat{\sigma}'_{u_t,t,\bar{\mathbf{x}}_t} + \eta' \hat{\sigma}'_{u,t,\bar{\mathbf{x}}_t}, \tag{6}$$

where the `prime` on v and σ denotes the fact that the values are computed similarly to equations (3) and (4), while only focusing on past-iterations concerning u or u'. Parameter $\eta' > 0$ controls the expected gap between clusters.

Finally, for each content $\mathbf{x}$ in the same content-cluster as $\bar{\mathbf{x}}_t$, we compute the neighborhood $N(\mathbf{x}) = \{u : |\hat{v}'_{u_t,t,\mathbf{x}} - \hat{v}'_{u,t,\mathbf{x}}| \le \eta' \hat{\sigma}'_{u_t,t,\mathbf{x}} + \eta' \hat{\sigma}'_{u,t,\mathbf{x}}\}$. We remove each edge $(\bar{\mathbf{x}}_t, \mathbf{x})$ such that this neighborhood differs from the one induced by the freshly updated users-graph.

4 CONCLUSIONS AND FUTURE WORKS

In this paper we demonstrate that collaborative bandits may be extended, through kernel trick, and get out of the linearity assumption.

REFERENCES

[1] Peter Auer. 2002. Using Confidence Bounds for Exploitation-Exploration trade-Offs. In *Journal of Machine Learning Research*. 397–422.

[2] Leonardo Cella. 2017. Modeling user behavior with evolving users and catalogs of evolving items. In *Extended Proceedings of the 25th UMAP conference*.

[3] Y. Chen, E. K. Garcia, R. M. Gupta, A. Rahimi, and L. Cazzanti. 2009. Similarity-based Classification: Concepts and Algorithms. In *Journal of Machine Learning Research*. 747–776.

[4] Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. 2011. Contextual Bandits with Linear Pauyoff Functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics AISTATS*.

[5] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A Contextual-Bandit Approach to personalized News Article Recommendation. In *WWW 2010*.

[6] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. July, 2016. Collaborative Filtering Bandits. In *SIGIR 16*. ACM, 176–185.

[7] J. Shawe-Taylor and N. Cristianini. 2004. Kernel Methods for Pattern Analysis. In *Cambridge University Press*.

[8] M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini. August, 2013. Finit-Time Analysis of Kernelised Contextual bandits. In *UAI'13 Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*. 654–663.