Dynamic Clustering of Contextual Multi-Armed Bandits

Introduction

In learning user preferences, recommender systems need to balance the trade-off between exploitation, by providing users with what they will most probably be interested in, and exploration, by providing users with something new so as to expand the systems knowledge. Under this framework, Multi Armed Bandits solve the exploration/exploitation problem altogether with the Upper Bound Confidence (UCB) algorithm.

The reinforcement learning algorithms implemented here on `MATLAB` have been described in the article of Nguyen and Lauw, *Dynamic Clustering of Contextual Multi-Armed Bandits*, and propose a new framework for recommender systems and Multi-Armed Bandits.

The Delicious dataset is used ; it can be found at `http://grouplens.org/datasets/hetrec-2011/`. It contains social networking, bookmarking, and tagging information from a set of 1867 users from the Delicious social bookmarking system, for 69226 URLs and 53388 tags.

The data has been preprocessed and two tables are attached to this file : a context matrix (each URL is associated with 25 features) as well as a `users` $\times$ `urls` matrix (users have assigned tags to specific URLs that they have bookmarked).

1 Contextual Multi Armed-Bandits

Multi-armed bandits have been shown to work well in various Web recommendation scenarios, such as advertisements, news articles, and comments.

1.1 Description

A contextual multi-armed bandits algorithm is an iterative algorithm that, at every step t, selects a random a user u_t that will choose an arm $a_t \in \mathcal{A}_t$, where $\mathcal{A}_t$ is the set of arms. Each arm is associated with a context vector x_{t,a_t}.

A popular framework for contextual bandits is **LinUCB**, which estimates the expected reward of each arm as a **linear regression** on the context vector : $r_{t,a_t} \propto w^T x_{t,a_t}$. Therefore, in LinUCB the selected arm a_t is the one maximizing the upper confidence bound $a_t = \underset{a_t}{\operatorname{argmax}}\ w^T x_{t,a_t} + \alpha\sqrt{x_{t,a_t}^T M^{-1} x_{t,a_t}}$. In many cases, it is advantageous to contextualize the bandit, such that the reward of an arm also depends on the *context* of a recommendation ; for instance, it can be the content of a Web page.

1.2 LinUCB-SIN

To build a SINgle bandit for all users - or, equivalently, a recommendation system that considers each user the same way -, an adapted contextual multi-armed bandits is LinUCB-SIN (LinUCB with a SINgle multi-armed bandit).

Here, at each iteration, the algorithm chooses the arm with the highest UCB, where the first term $\bar{w}^T x_{t,a}$ is a convenient representation of the expected reward for the whole set of users, $\bar{w}$ being the regression coefficient to be learnt, and the second term is the confidence bound ($\bar{M}$ being a convenient representation of the covariance of the regression coefficient $\bar{w}$). The parameter α gives weight to exploration.

1.3 LinUCB-IND

It is also possible to train a bandit for every INDividual user : the algorithm is called LinUCB-IND. While LinUCB-SIN benefits from training instances and the way they allow to describe the whole population, LinUCB-IND allows to consider each user independently, and to build for this user a customized bandit. The algorithm can be implemented as follows :

Algorithm 1: LinUCB-SIN

Input: Context vectors $\{x_{t,a}\}\forall t, a$.

Output: At iteration t, recommended arm a_t for u_t

Set $b_u = 0 \in \mathbb{R}^d$, $M_u = I \in \mathbb{R}^{d\times d}$ for all users $u = 1, \ldots, N$.

Compute the coefficient $\bar{w}$ from the set of users C:

  $\bar{M} = I + \sum_{u'\in C}(M_{u'} - I)$

  $\bar{b} = \sum_{u'\in C} b_{u'}$

  $\bar{w} = \bar{M}^{-1}\bar{b}$

for *iteration* $t = 1, \ldots, T$ **to** B **do**

  Select a user u_t.

  Observe the contexts of arms $\{x_{t,a}\}$,$\forall a \in \mathcal{A}_t$.

  Find the arm a_t with the highest UCB, i.e.,

$$a_t = \underset{a\in\mathcal{A}_t}{\operatorname{argmax}} \left(\bar{w}^T x_{t,a} + \alpha\sqrt{x_{t,a}^T \bar{M}^{-1} x_{t,a} \log(t+1)} \right) \tag{1}$$

  Observe the reward r_{t,a_t} from recommending a_t.

  Let $\tilde{x}_t = x_{t,a_t}$.

  Update the user u_t's parameters :

    $M_{u_t} = M_{u_t} + \tilde{x}_t \tilde{x}_t^T$

    $b_{u_t} = b_{u_t} + r_{t,a_t}\tilde{x}_t$

    $w_{u_t} = M_{u_t}^{-1} b_{u_t}$

  Re-compute coefficients $\bar{w}$, $\bar{b}$ and $\bar{M}$ as above.

Algorithm 2: LinUCB-IND

Input: Context vectors $\{x_{t,a}\}\forall t, a$.

Output: At iteration t, recommended arm a_t for u_t

Set $b_u = 0 \in \mathbb{R}^d$, $M_u = I \in \mathbb{R}^{d\times d}$ for all users $u = 1, \ldots, N$.

for *iteration* $t = 1, \ldots, T$ **to** B **do**

  Select a user u_t.

  Observe the contexts of arms $\{x_{t,a}\}$,$\forall a \in \mathcal{A}_t$.

  Find the arm a_t with the highest UCB, i.e.,

$$a_t = \underset{a\in\mathcal{A}_t}{\operatorname{argmax}} \left(w_{u_t}^T x_{t,a} + \alpha\sqrt{x_{t,a}^T M_{u_t}^{-1} x_{t,a} \log(t+1)} \right) \tag{2}$$

  Observe the reward r_{t,a_t} from recommending a_t.

  Let $\tilde{x}_t = x_{t,a_t}$.

  Update the user u_t's parameters :

    $M_{u_t} = M_{u_t} + \tilde{x}_t \tilde{x}_t^T$

    $b_{u_t} = b_{u_t} + r_{t,a_t}\tilde{x}_t$

    $w_{u_t} = M_{u_t}^{-1} b_{u_t}$

1.4 Dyn-UCB : Dynamic Clustering of users

In the article of Nguyen and Lauwn, authors hypothesized that users sharing the same kinds of preferences might be splitted into the same clusters. Indeed, making a single cluster for the whole population might not be as accurate as adapting the clusters to similar users. Moreover, individual bandits adapted to every user might take too long to learn from every user, and would greatly benefit from the reward experience of similar users. Therefore, making groups for similar users will customize the bandits to the available population. The **dynamic clustering approach** comes from the fact that users can switch from one cluster to another, regarding their reward experience (captured through the w_u vector), and their distances to the users clusters' average users experiences (captured through $\bar{w}_k$ for cluster k). The optimal number of clusters K depends on the dataset and can be optimized during implementation.

The algorithm can be implemented as follows in the next page.

Algorithm 3: DynUCB

Input: The number of clusters K, context vectors $\{x_{t,a}\}\forall t, a$.
Output: At iteration t, recommended arm a_t for u_t

Set $b_u = 0 \in \mathbb{R}^d$, $M_u = I \in \mathbb{R}^{d\times d}$ for all users $u = 1, \ldots, N$.
Randomly assign users to K clusters $\{C_k\}_{k=1}^K$.
Compute the coefficient $\bar{w}_k$ for each cluster C_k:
    $\bar{M}_k = I + \sum_{u'\in C_k}(M_{u'} - I)$
    $\bar{b}_k = \sum_{u'\in C_k} b_{u'}$
    $\bar{w}_k = \bar{M}_k^{-1}\bar{b}_k$
for *iteration* $t = 1, \ldots, T$ **to** B **do**
    Select a user u_t, and its current cluster $C_k \ni u_t$.
    Observe the contexts of arms $\{x_{t,a}\}$,$\forall a \in \mathcal{A}_t$.
    Find the arm a_t with the highest UCB, i.e.,

$$a_t = \underset{a\in\mathcal{A}_t}{\operatorname{argmax}} \left(\bar{w}_k^T x_{t,a} + \alpha\sqrt{x_{t,a}^T \bar{M}_k^{-1} x_{t,a} \log(t+1)} \right) \tag{3}$$

    Observe the reward r_{t,a_t} from recommending a_t.
    Let $\tilde{x}_t = x_{t,a_t}$.
    Update the user u_t's parameters :
        $M_{u_t} = M_{u_t} + \tilde{x}_t\tilde{x}_t^T$
        $b_{u_t} = b_{u_t} + r_{t,a_t}\tilde{x}_t$
        $w_{u_t} = M_{u_t}^{-1} b_{u_t}$

    Re-assign the user u_t to the closest cluster $C_{k'}$:
        $k' = \underset{k'=1,\ldots,K}{\operatorname{argmin}} \|w_{u_t} - \tilde{w}_{k'}\|$
        If $k' \neq k$, move u_t from C_k to $C_{k'}$
        Re-compute coefficients $\bar{w}_k$ and $\bar{w}_{k'}$ as above.