Data Mining: Learning from Large Data Sets - Fall Semester 2015

luchen@student.ethz.ch
zhhan@student.ethz.ch
myhshirley@student.ethz.ch

December 24, 2015

Explore-Exploit Tradeoffs in Recommendation Systems

The goal of this task is to learn a policy that explores and exploits among available choice in order to learn user preferences and recommend news articles to users. We apply the LinUCB algorithm in [1], as shown in Figure 1. Following are the details of the three functions, i.e. *set_articles*, *reccomend* and *update*, in the code file *policy.py*.

Algorithm 1 LinUCB with disjoint linear models.

Inputs: $\alpha \in \mathbb{R}_+$
for $t = 1, 2, 3, \ldots, T$ **do**
  Observe features of all arms $a \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
  for all $a \in \mathcal{A}_t$ **do**
    if a is new **then**
      $\mathbf{A}_a \leftarrow \mathbf{I}_d$ (d-dimensional identity matrix)
      $\mathbf{b}_a \leftarrow \mathbf{0}_{d \times 1}$ (d-dimensional zero vector)
    end if
    $\hat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$
    $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha\sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1}\mathbf{x}_{t,a}}$
  end for
  Choose arm $a_t = \arg\max_{a \in \mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_t
  $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^\top$
  $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t\mathbf{x}_{t,a_t}$
end for

Figure 1: Algorithm 1 LinUCB with disjoint linear models in [1].

set_articles When the *evaluator* reads all the articles, it will call this function to store the information of these articles into a dictionary, and to initiate the matrix A and vector b corresponding to each article.

reccomend Each time the *evaluator* reads a line in the log, it will send the relative information *time, user_features, articles* to this function. Note that the third argument *articles* is a list of articles available for selection to that user. Then this function will compute the UCB of each available article and recommend the one with the highest bound.

update If the choice made by *reccomend* matches the one displayed to the user in the log file, this function will be called. It will update the matrix A and vector b associated to this article by the user feature and the observed payoff.

Contribution

Each of us worked out a solution and we put together our codes to get the final solution. We also tried different values of the parameter ALPHA until we find a solution which reaches the hardline.

References

[1] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In WWW-10, Raleigh, NC, April, 2010.