Application of contextual bandit algorithms on referral data

Sandeep Kumar Gangarapu

November 15, 2018

We apply LinUCB algorithm on referral data which was the output of randomized field experiment.

In order to simulate contextual bandit procedure and compare it with A/B testing, we do the following trick.

1. Shuffle the data
2. Randomly sample one feature vector(person)
3. Ask the algorithm where the person should be allocated
4. Peek at the actual group and Check whether the suggested group is same as the actual group.
5. If yes, we call this an optimal allocation and simulate pulling the lever by observing the reward(peeking at the value of firm gain)
6. If no, discard the datapoint and move on to the next one

The above process of filtering data points that follow the orders of contextual bandit algorithm simulates the real process of contextual bandits.

We could also store all the discarded points and call them suoptimal and simulate pulling the lever by observing the reward(peeking at the value of firm gain)

We expect the contextual bandit (optimal) allocation and subsequent aggregate reward to be better than random allocation and suboptimal

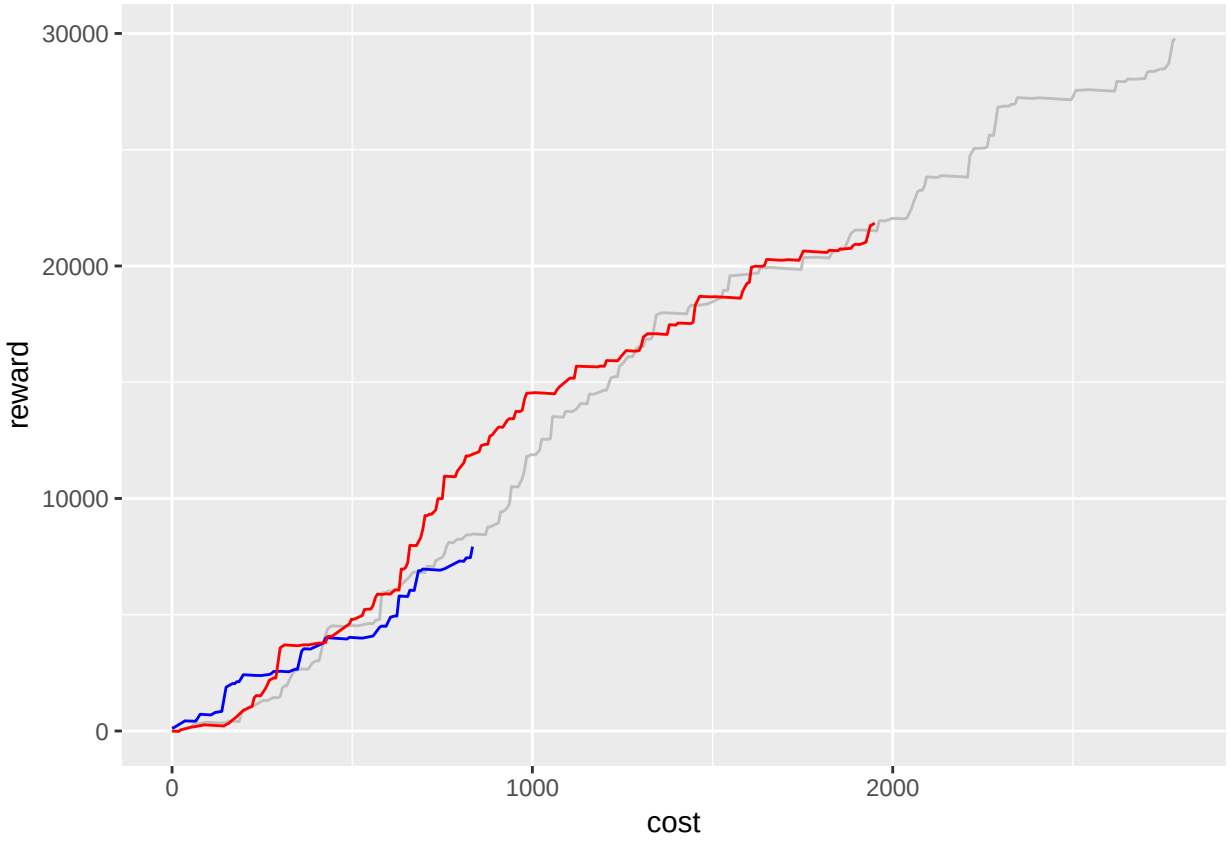
The below graphs demonstrate whether this is the case

We observe this for different values of alpha which is a parameter that gives weight to the upper confidence bound. The higher the value of alpha the more exploration the algorithm does.

for all the graphs red line represents suboptimal, grey line represents random and green line represents optimal

alpha = 1

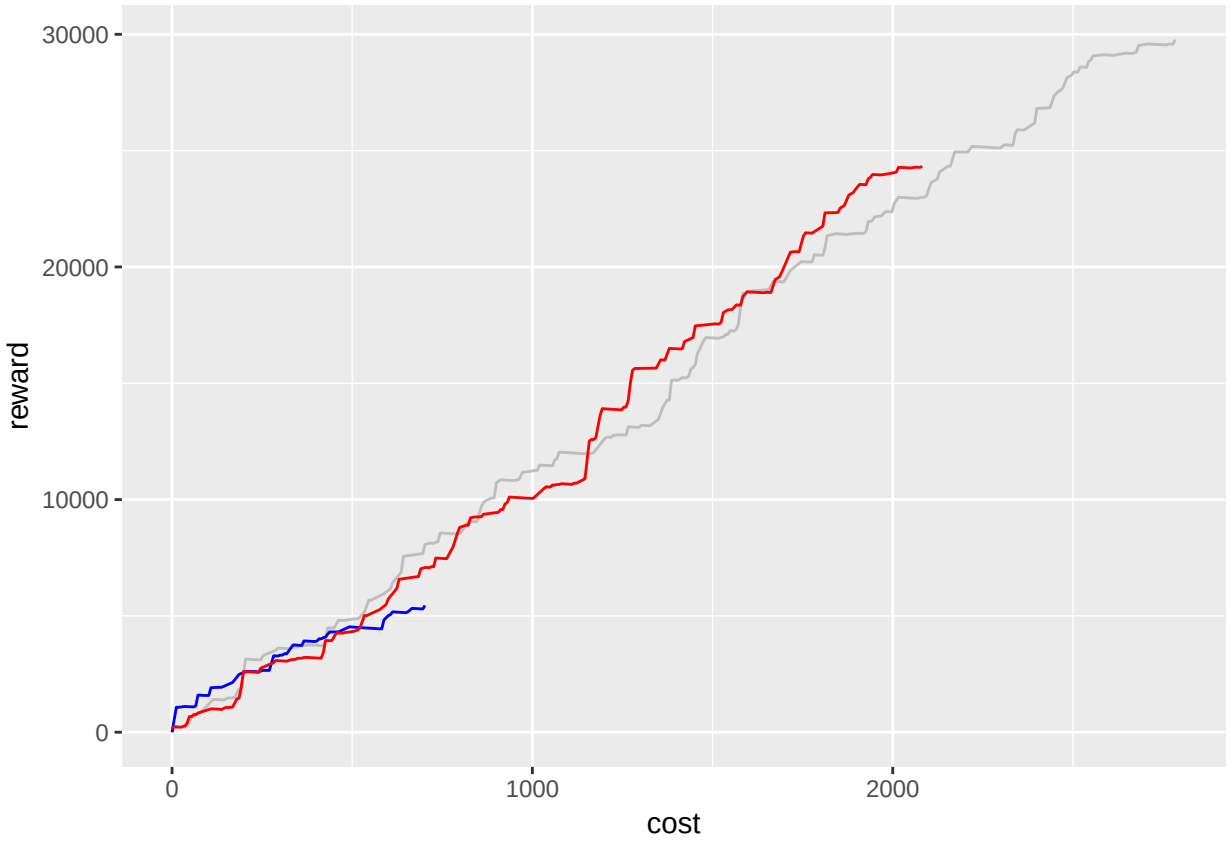
iteration = 1



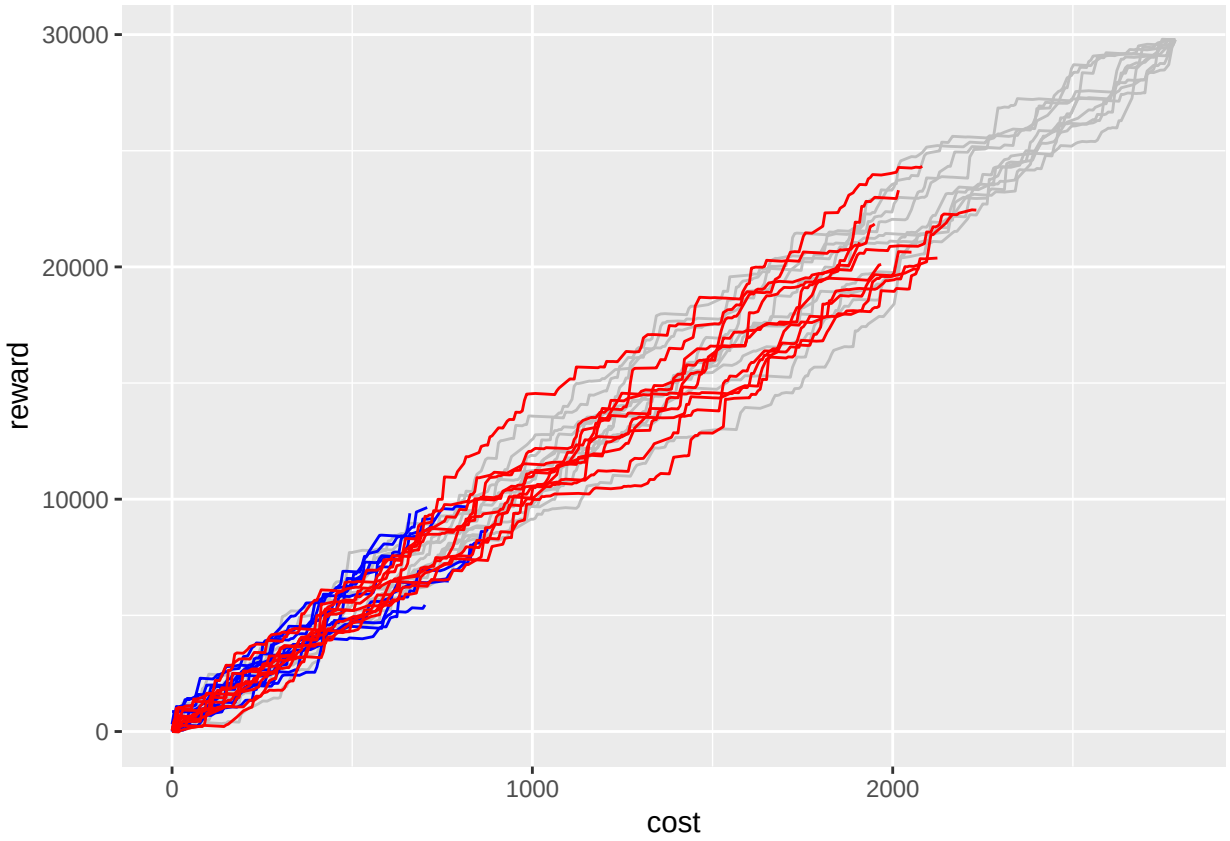
We suspect this may be because of random order of arrival of persons (random sampling of datapoints), So we check again

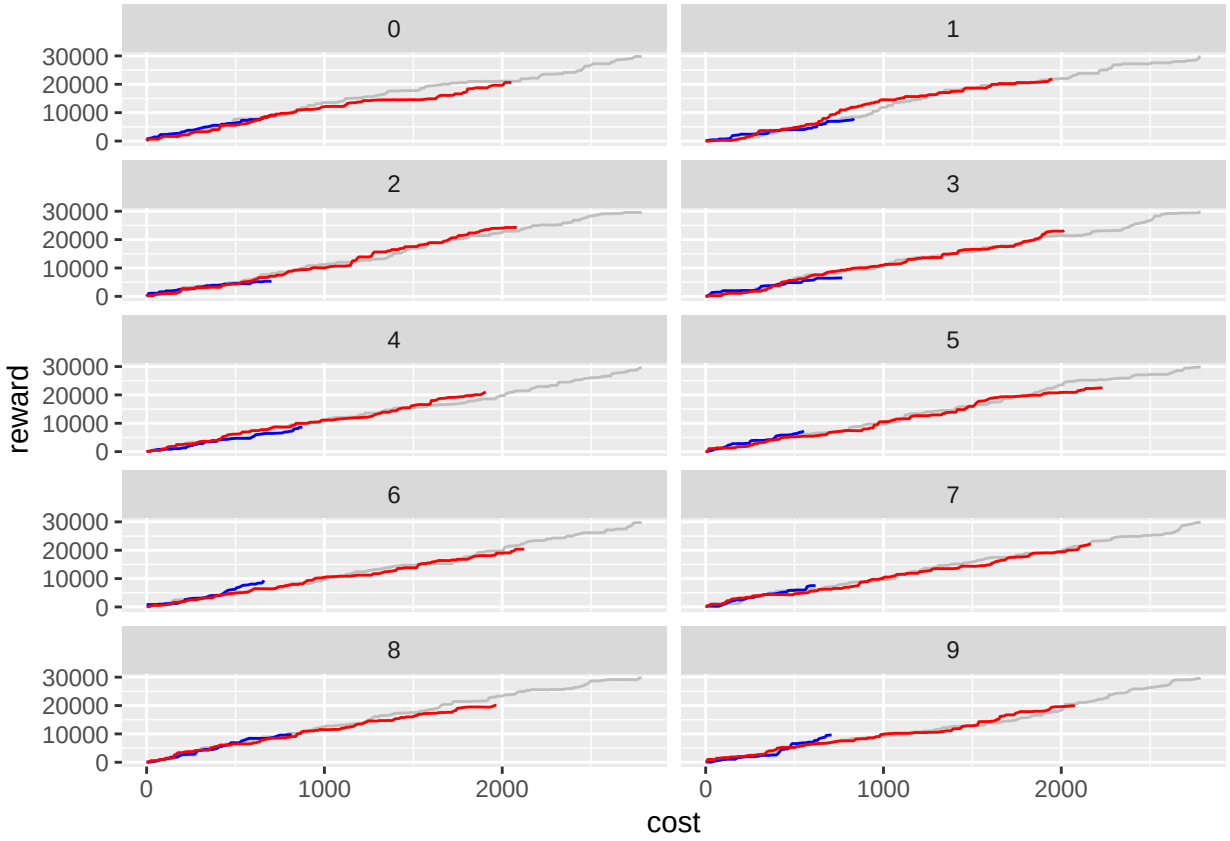
alpha = 1

iteration = 2



Just to be sure, we do it 10 different times

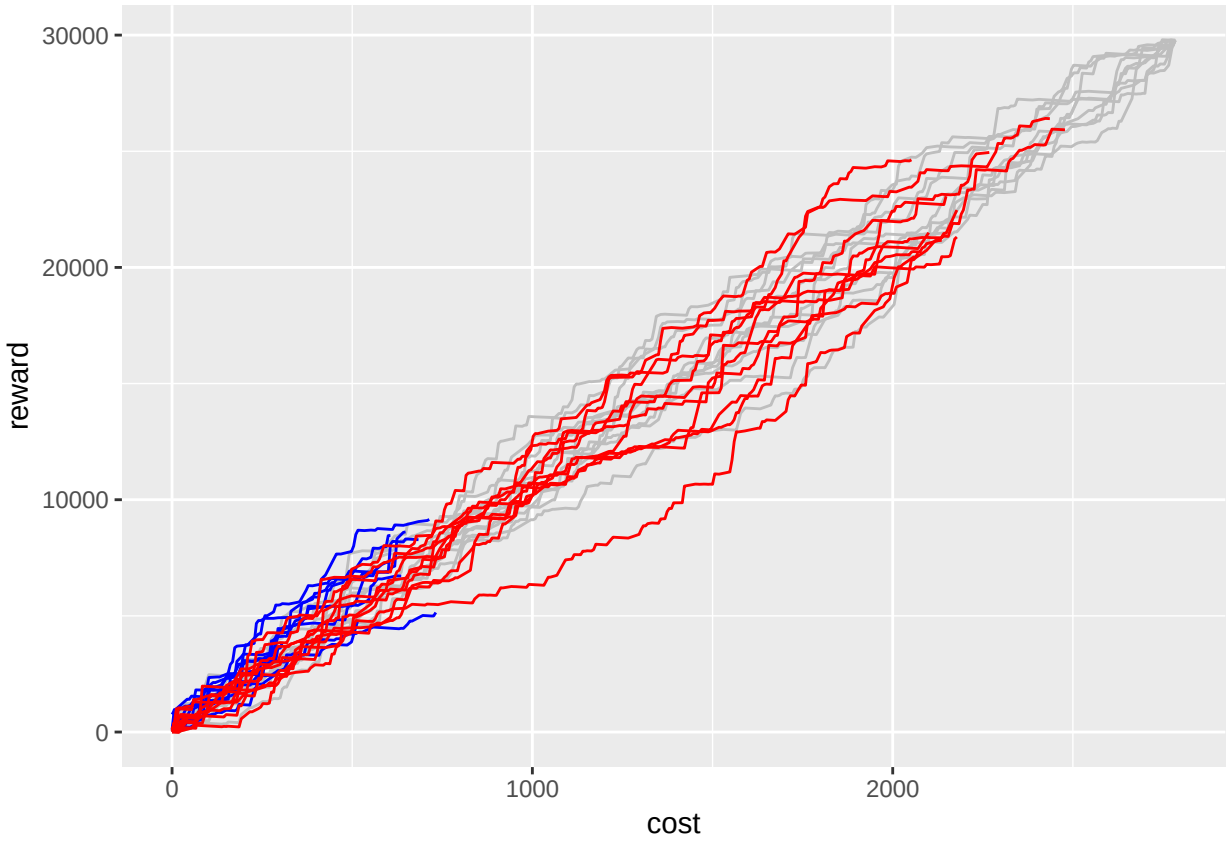

30000
20000
10000
0
reward
0
1000
2000
cost

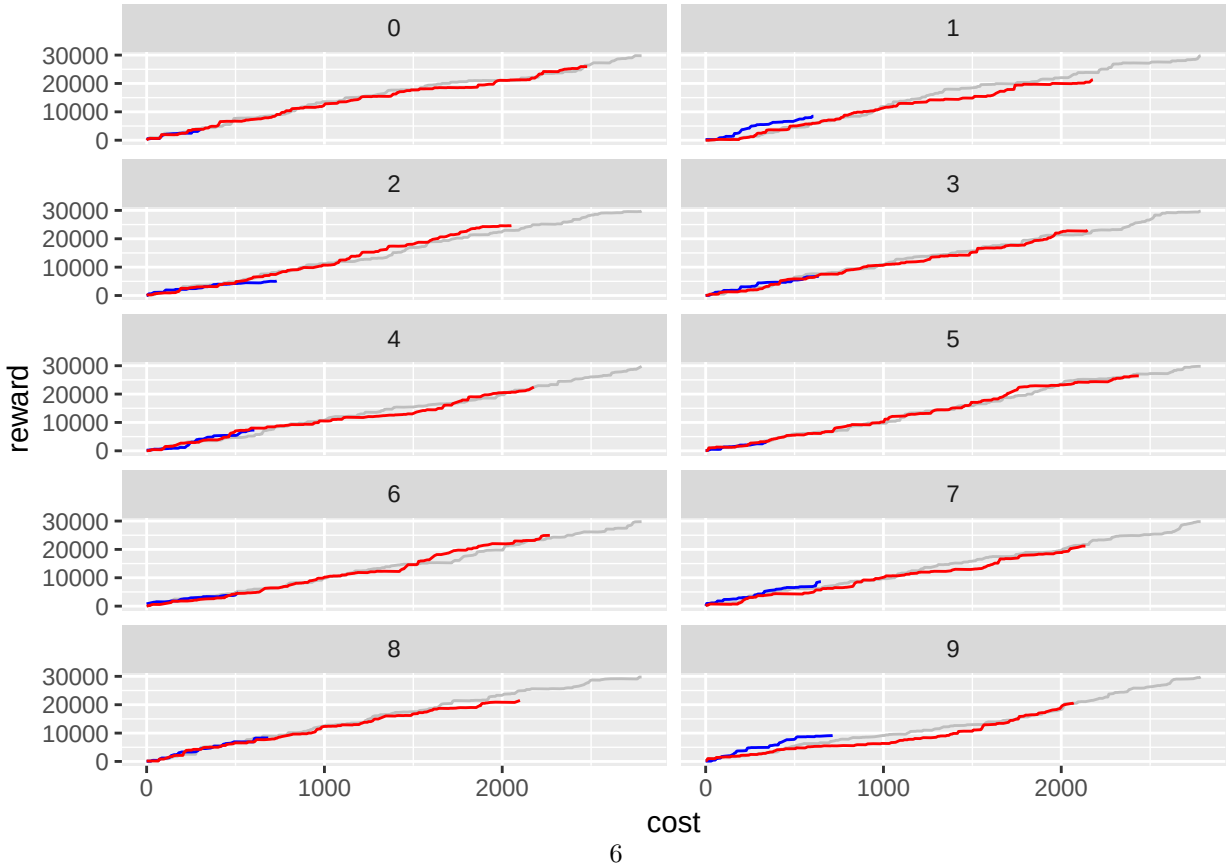


Its consistent but some iterations have improper order

We can check it for other alphas

alpha=10


30000
20000
10000
0
reward
0
1000
2000
cost


0
1
2
3
4
5
6
7
8
9
30000
20000
10000
0
reward
0
1000
2000
cost

This could be because of unbalanced treatment classes. There are more people allocated to control group than other treatment groups.

Var1	Freq
0	9989
1	29960
2	29968
3	29963

We check the behavior for both.

Upsampling treatment groups

Before

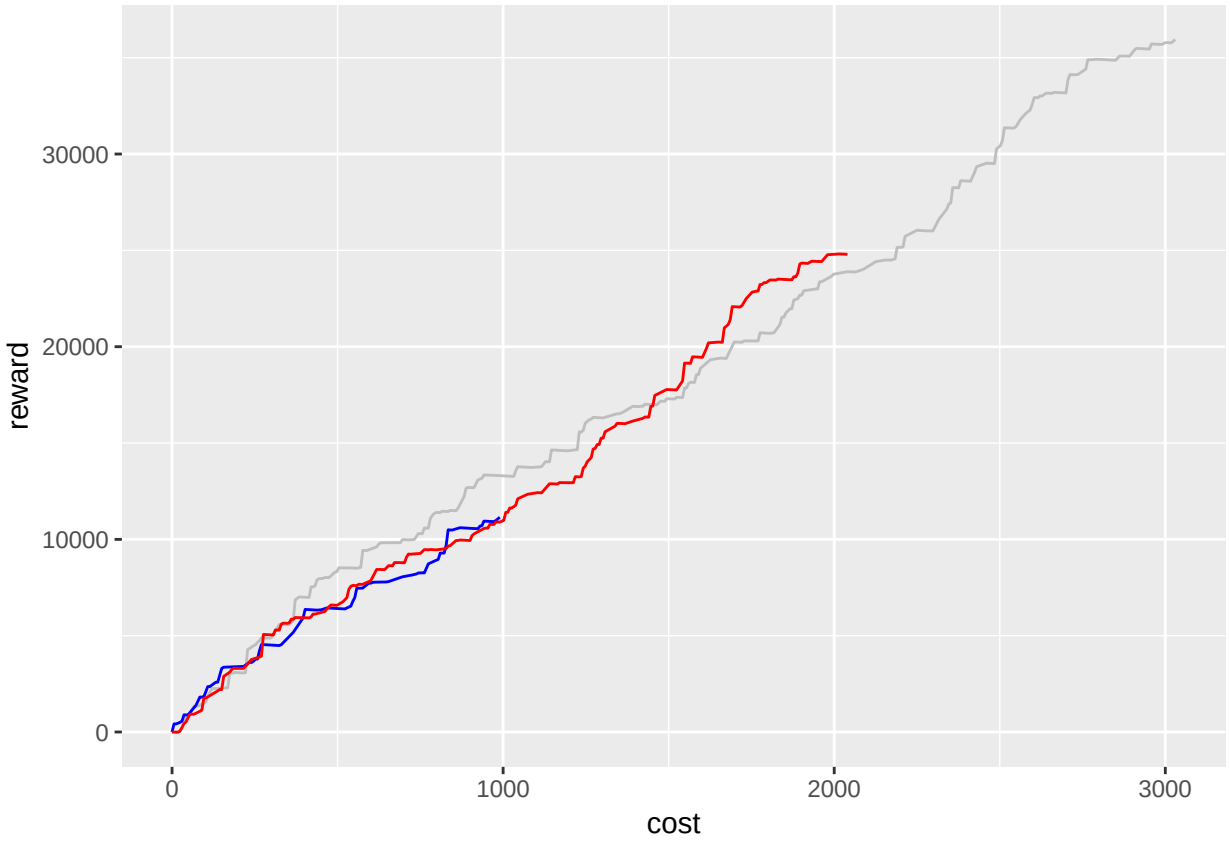
Var1	Freq
0	9989
1	29960
2	29968
3	29963

After

Var1	Freq
0	29968
1	29968
2	29968
3	29968

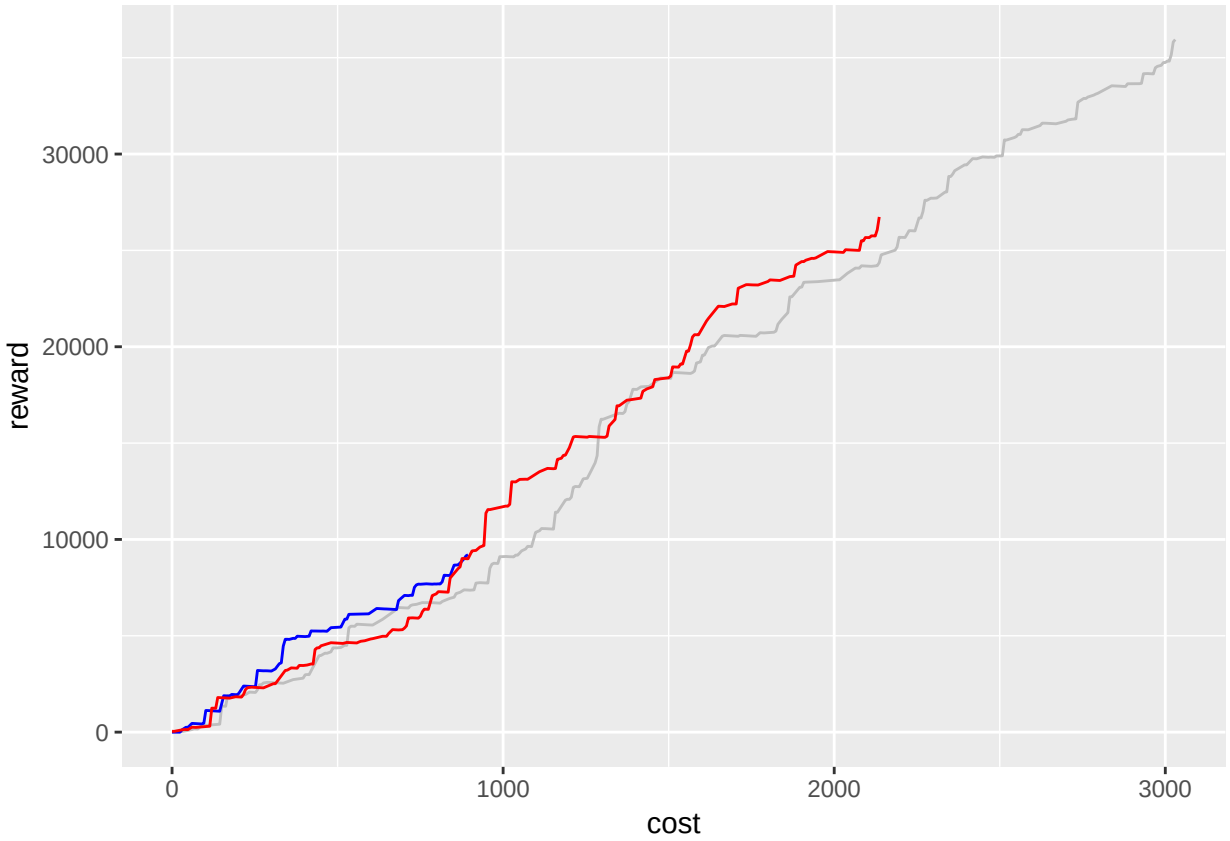
alpha = 1

iteration = 1

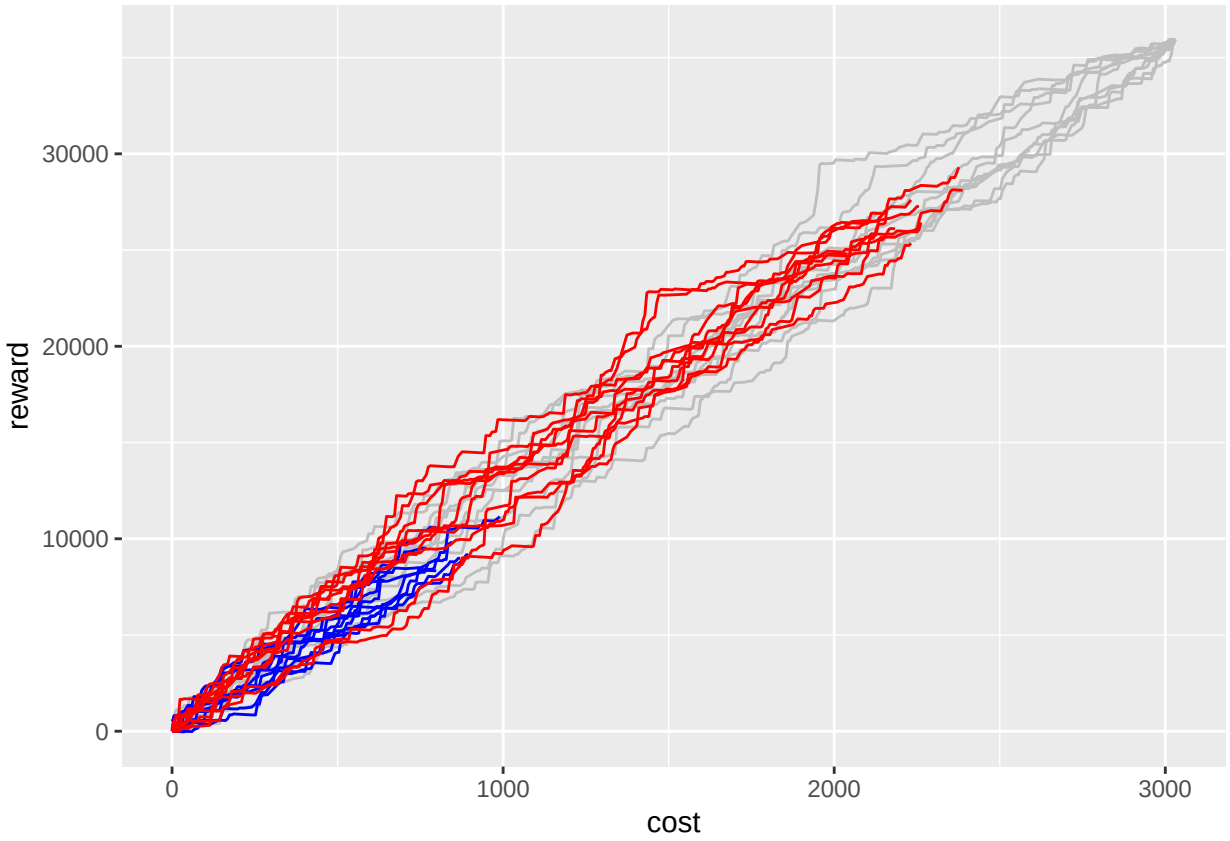


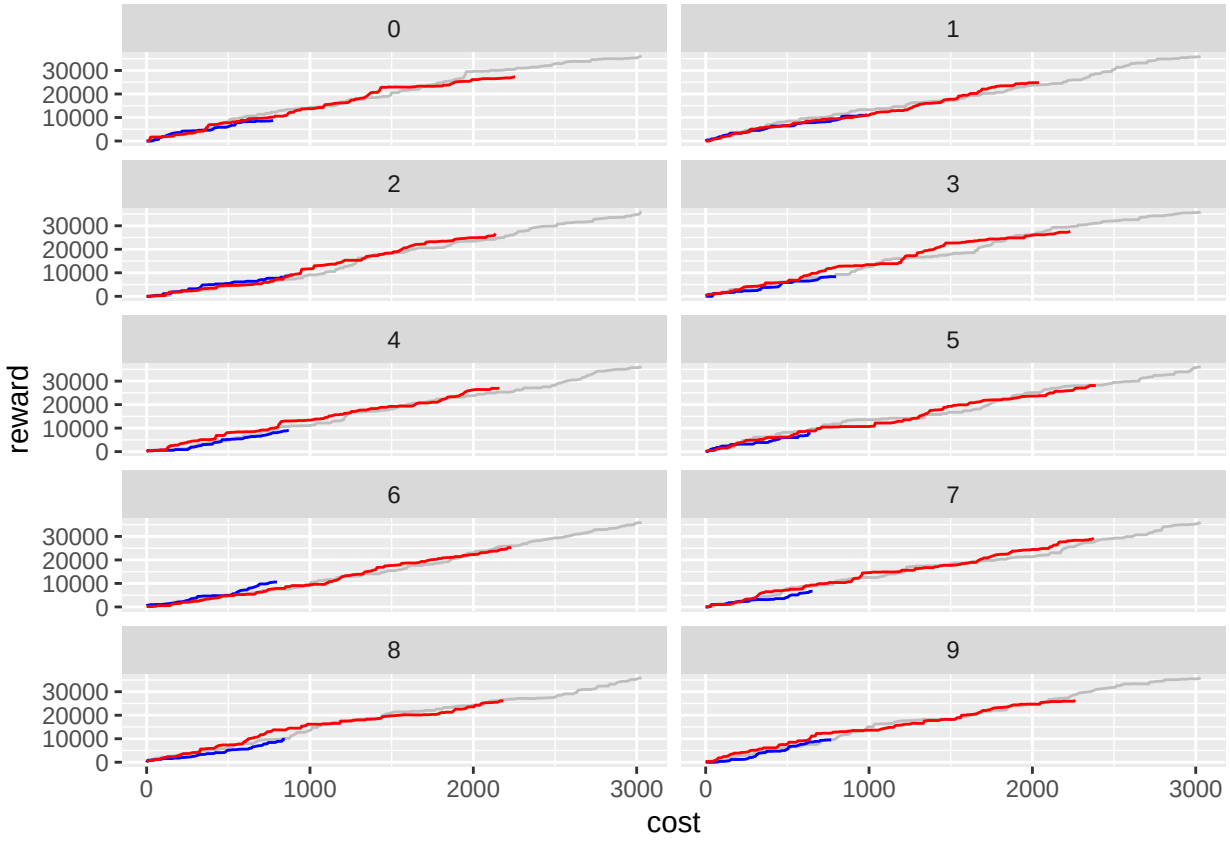
alpha = 1

iteration = 2



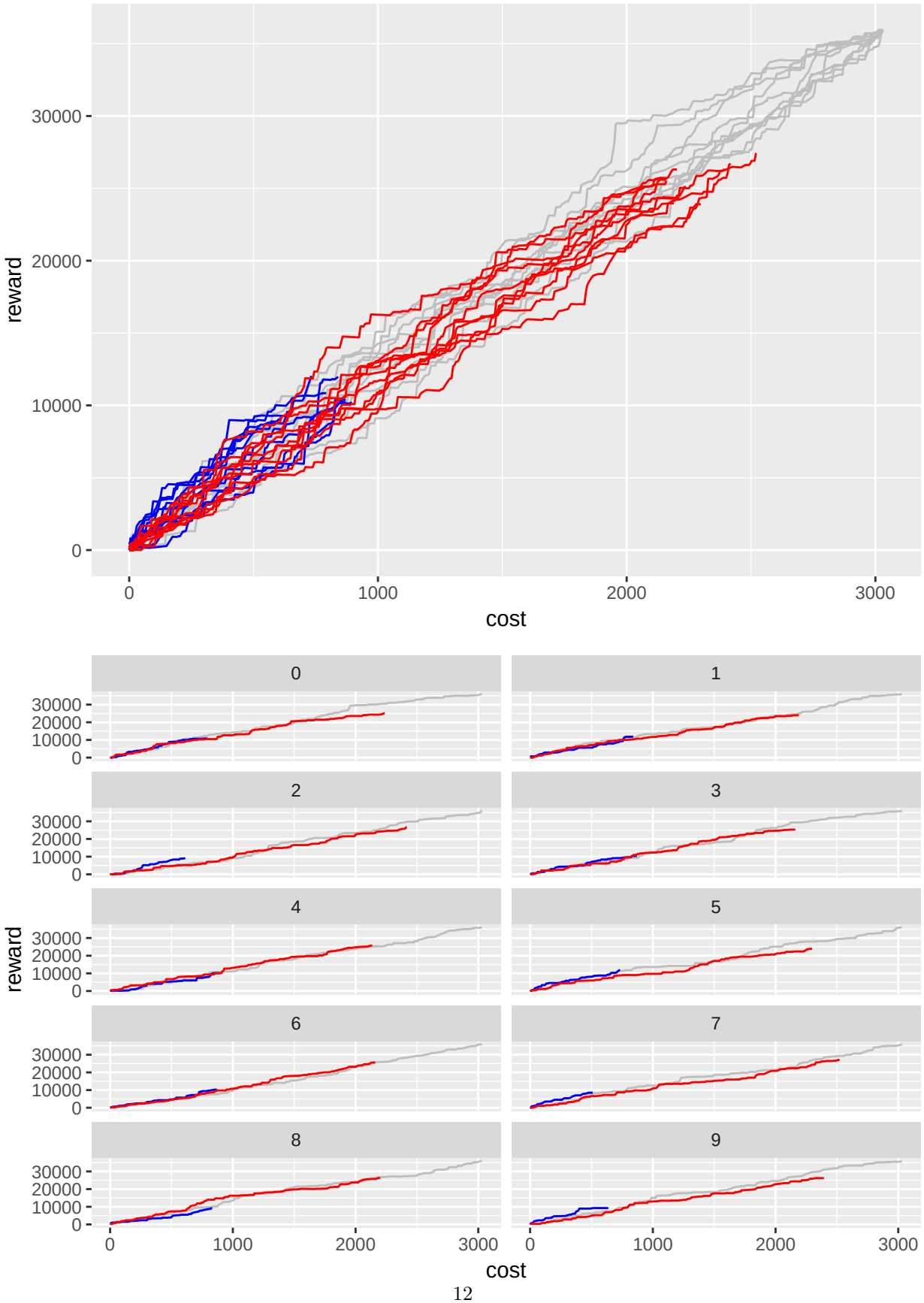
Just to be sure, we do it 10 different times


30000
20000
10000
0
reward
0
1000
2000
3000
cost



We can check it for other alphas

alpha=15



There seem to be no significant difference overall.

This could be attributed to sprcity of data