TS_Sampling_demo

Xiao Chen

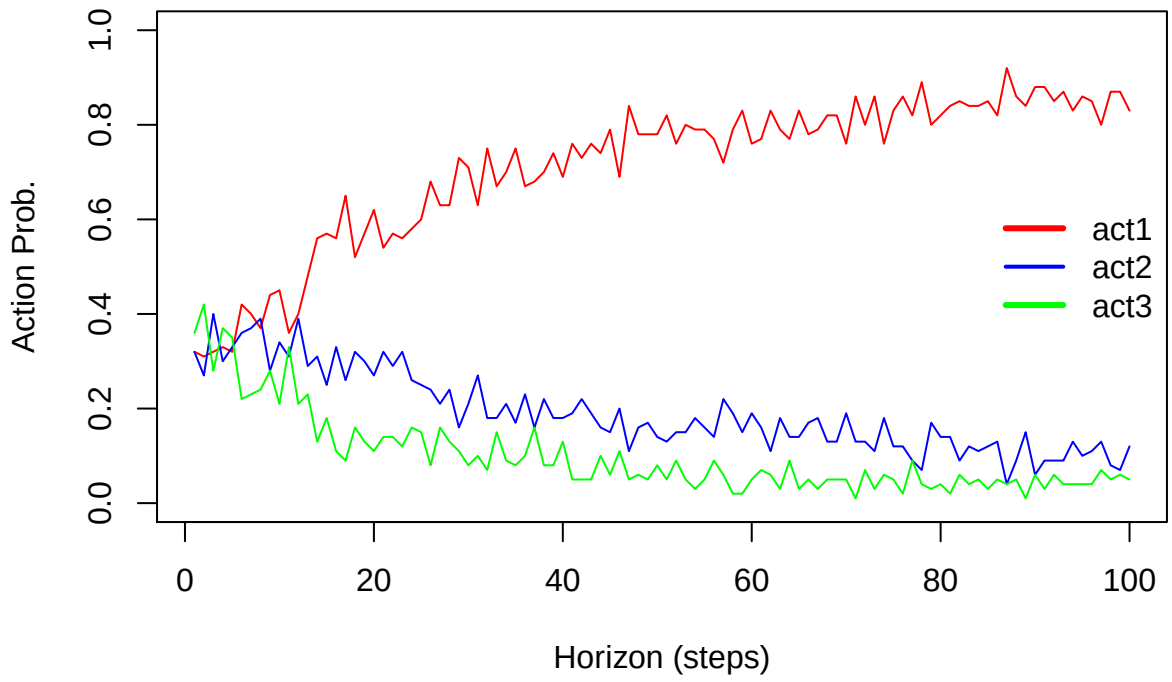
Bernulli Bandit Learning

This is the document performing ε-greedy and Thompson Sampling

Thompson Sampling

```
sim_run = function(iter_n=1, Horizon = 1000  ){

    histT=data.frame(row.names=paste("t", c(1:Horizon), sep = '' ) )
    paraT = data.frame(alpha= rep(1,3), beta = rep(1,3) )
    K = nrow(paraT)
    X = data.frame(x1=rep(0,Horizon),
                   x2=rep(0,Horizon),
                   x3=rep(0,Horizon),
                   opt_act=rep(0, Horizon) )

    # actionProb = data.frame(act1=rep(0,Horizon),
    #                 act2=rep(0,Horizon),
    #                 act3=rep(0,Horizon) )
    for (t in 1:Horizon){
      theta =  sapply(c(1:K), function(k) {rbeta(n=1, paraT[k,1], paraT[k,2])} )
      # print(theta)
      idx = which.max(theta)
      X[t, "opt_act"] = idx
      X[t, 1:3] = theta
      # actionProb[t, 1:3] = theta # / sum(theta)
      if (idx == 1){
        r = 0.9
        # r = X[t, "x1"]
      }else if (idx==2){
        r = 0.8
        # r = X[t, "x2"]
      }else if (idx ==3) {
        r = 0.7
        # r = X[t, "x3"]
      }

      paraT$alpha[idx] = paraT$alpha[idx] + r
      paraT$beta[idx] = paraT$beta[idx] + 1 - r
    }
    histT[paste("sim", iter_n, sep = '')] = X["opt_act"]

  return(histT)
}

library(parallel)
output = mclapply( 1:100, function(i) {sim_run(iter_n = i, Horizon = 100)}  )
output = as.matrix(data.frame(output))
N_cols = ncol(output)
N_rows = nrow(output)
tableFreq = data.frame(act1=rep(0, N_rows ),
                       act2=rep(0, N_rows ),
                       act3=rep(0, N_rows ))
# calculate the empirical results
```

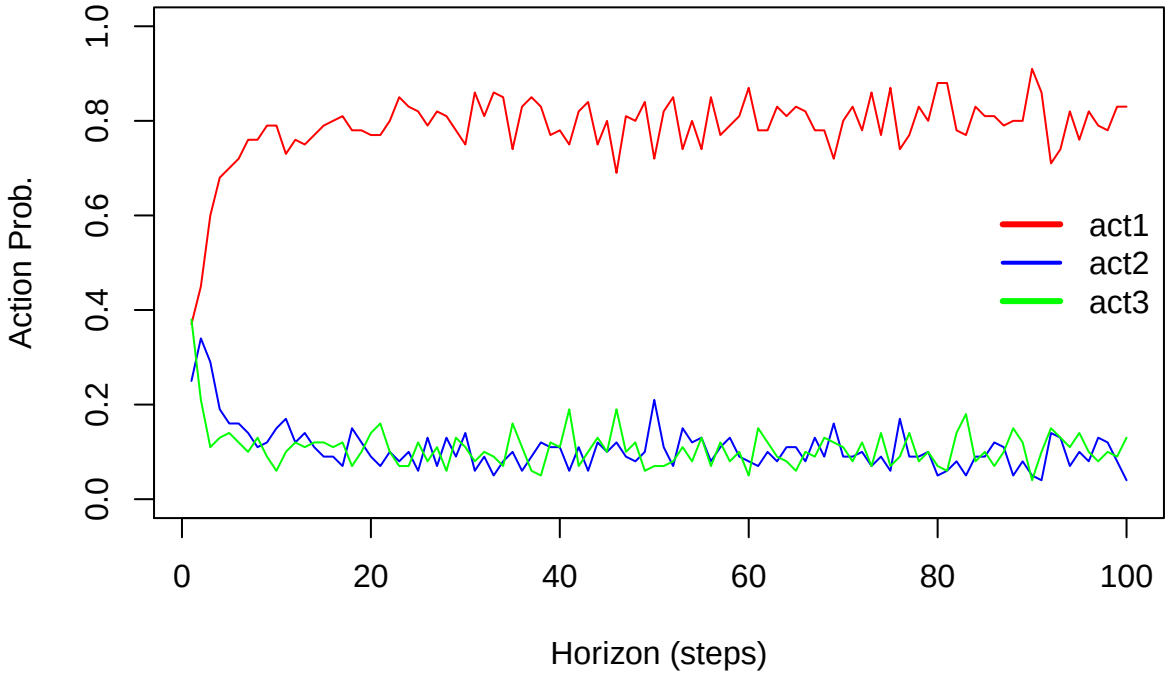
```
for (k in 1:nrow(output) ){
  tableFreq[k, ] = (sapply(c(1:3), function(x) {return(length(which(output[k, ] == x))) } )) / N_cols
}

# library(ggplot2)
# slimT = reshape2::melt(tableFreq)
# ggplot(slimT, aes(x=1:3000, y=value, group=variable ))
plot.new()
plot(x=seq(1:nrow(tableFreq)), y=tableFreq$act1, type="l", col="red", ylim = c(0,1),
     xlab = "Horizon (steps)", ylab = "Action Prob." )
lines(x=seq(1:nrow(tableFreq)), y=tableFreq$act2, col="blue" )
lines(x=seq(1:nrow(tableFreq)), y=tableFreq$act3, col="green" )
legend("right", c("act1", "act2", "act3"), col = c("red", "blue", "green"),
       lty = c(1, 1, 1), lwd = c(3, 2), bty = "n" )
```



ε Greedy Algorithm

```
sim_run_greedy = function(iter_n=1, Horizon = 1000, eps=0.1, ...  ){

    histT=data.frame(row.names=paste("t", c(1:Horizon), sep = '' ) )
    paraT = data.frame(alpha= rep(1,3), beta = rep(1,3) )
    K = nrow(paraT)
    X = data.frame(x1=rep(0,Horizon),
                   x2=rep(0,Horizon),
                   x3=rep(0,Horizon),
                   opt_act=rep(0, Horizon) )

    # actionProb = data.frame(act1=rep(0,Horizon),
    #                 act2=rep(0,Horizon),
    #                 act3=rep(0,Horizon) )
    for (t in 1:Horizon){
      theta =  sapply(c(1:K), function(k) {  return(paraT[k,1]/(paraT[k,1] + paraT[k,2])) } )
      # print(theta)
      rnd.flag = runif(1, min=0, max=1)
      idx.max = sample(which(theta==max(theta)), 1)
      if (rnd.flag > eps ) {
        idx = idx.max
```

```
      }else{
        act.ids = c(1:length(theta))
        idx = sample(act.ids[-idx.max], 1)
      }

      X[t, "opt_act"] = idx
      X[t, 1:3] = theta
      # actionProb[t, 1:3] = theta # / sum(theta)
      if (idx == 1){
        r = 0.9
        # r = X[t, "x1"]
      }else if (idx==2){
        r = 0.8
        # r = X[t, "x2"]
      }else if (idx ==3) {
        r = 0.7
        # r = X[t, "x3"]
      }

      paraT$alpha[idx] = paraT$alpha[idx] + r
      paraT$beta[idx] = paraT$beta[idx] + 1 - r
    }
    histT[paste("sim", iter_n, sep = '')] = X["opt_act"]

  return(histT)
}

library(parallel)
output = mclapply( 1:100, function(i) {sim_run_greedy(iter_n = i, Horizon = 100, eps=0.2)}  )
output = as.matrix(data.frame(output))
N_cols = ncol(output)
N_rows = nrow(output)
tableFreq = data.frame(act1=rep(0, N_rows ),
                       act2=rep(0, N_rows ),
                       act3=rep(0, N_rows ))
# calculate the empirical results
for (k in 1:nrow(output) ){
  tableFreq[k, ] = (sapply(c(1:3), function(x) {return(length(which(output[k, ] == x))) } )) / N_cols
}

# library(ggplot2)
# slimT = reshape2::melt(tableFreq)
# ggplot(slimT, aes(x=1:3000, y=value, group=variable ))
plot.new()
plot(x=seq(1:nrow(tableFreq)), y=tableFreq$act1, type="l", col="red", ylim = c(0,1),
     xlab = "Horizon (steps)", ylab = "Action Prob." )
lines(x=seq(1:nrow(tableFreq)), y=tableFreq$act2, col="blue" )
lines(x=seq(1:nrow(tableFreq)), y=tableFreq$act3, col="green" )
legend("right", c("act1", "act2", "act3"), col = c("red", "blue", "green"),
       lty = c(1, 1, 1), lwd = c(3, 2), bty = "n" )
```


Action Prob.
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Horizon (steps)
act1
act2
act3