Online Sparse Bandits

David L. Saint-Pierre, Quentin Louveaux, Olivier Teytaud

Online Sparse bandit

David L. St-Pierre[1], Quentin Louveaux[1], and Olivier Teytaud[2,3]

[1] Department of Electrical Engineering and Computer Science, Faculty of Engineering, Liège University, Belgium
[2] TAO (Inria, Lri, Univ. Paris-Sud, UMR CNRS 8623), France
[3] OASE Lab, National University of Tainan, Taiwan

What is a zero-sum Matrix Game (MG) ?

M = KxK matrix, coefficients in [0,1]
Player 1 chooses i in {1,2,3,...,K}
Player 2 chooses j in {1,2,3,...,K}
Player 1 gets reward Mij
Player 2 gets reward 1-Mij

Is solving a MG hard ?

= polynomial time (by linear programming)
best known coef 3.5

How to find approximate solutions ?

Grigoriadis & Khachiyan or Exp3 or Inf:
time O(Klog K / ε2) with proba ½
(Auer et al, Audibert et al, Grigoriadis et al)

Sparse versions ?

Very often, (x*,y*) is very sparse; plenty of 0's.
How to benefit from this ?
Algo. in Flory et al.:

1. Approximate solving by t iterations of EXP3
2. Remove small components (keep only components ≥ (max (tx))$^{(4/5)}$ / t)
3. Re-normalize
   ==> no proof

We propose the following online version ==>

(EXP3 recalled below...)

Algorithm 1 EXP3 algorithm for iteration t with K arms.

Initialise $\forall i, p(i) = \frac{1}{K}, n(i) = 0, S(i) = 0$; $t = 0$
while t < T **do**
  Arm i is chosen with probability $p(i)$
  n(i) ← n(i)+1
  Receive reward r
  t ← t+1
  S_i modified by the update formula $S_i \leftarrow S_i + r/p(i)$ (and S_j for $j \neq i$ is not modified).
  $\forall i, p(i) = 1/(K\sqrt{t}) + (1-1/\sqrt{t}) \times \exp(S_i/\sqrt{t})/\sum_j \exp(S_j/\sqrt{t})$
end while
return n

What means "solving" a MG ?

Strategy x: probability distribution on {1,2,3,....,K}
Strategy y: probability distribution on {1,2,3,...,K}
Expected Reward:
R(x,y)=
E Mij | (i,j)~x,y
Nash equilibrium:
(x*,y*) = Nash
<==> for all (x,y)
R(x,y*) ≥ R(x*,y*) ≥ R(x*,y)

Approximate solving ?

(x*,y*) ε-approximate Nash equilibrium if
for all (x,y), R(x,y*)+ε ≥ R(x*,y*) ≥ R(x*,y)-ε

So what ?

There is a *offline* solution
(i.e. sparsity used at the end).
Can we use it online ?

Algorithm 3 onEXP3, an online EXP3 algorithm with a cut solely based on T.

Initialise $\forall i, p(i) = \frac{1}{K}, n(i) = 0, S(i) = 0$; $t = 0$
while t < T **do**
  Arm i is chosen with probability $p(i)$
  n(i) ← n(i)+1
  t ← t+1
  Receive reward r
  S_i modified by the update formula $S_i \leftarrow S_i + r/p(i)$
  $\forall i, p(i) = 1/(K\sqrt{t}) + (1-1/\sqrt{t}) \times \exp(S_i/\sqrt{t})/\sum_j \exp(S_j/\sqrt{t})$
  if $x_i > \lceil \frac{t}{K} \rceil$ **and** $x_i < (b_1 \times T^{\delta} \times (\frac{t}{T})^{\beta})$ **then**
    Remove arm i
  end if
  if every arm has been pruned **then**
    Use plain EXP3
  end if
  Renormalize: $p = p/\sum_i p(i)$
end while
Execute the truncation TEXP3 as presented in 2.3
return n

Conclusions ?

(i) it works (see numbers in paper) (ii) theory missing (iii) better (parameter-free ?) versions

<== application: Urban Rivals
(free, you can test!)
Next application: Pokemon ==>