Bandits on stochastic block-model graphs

Arthur Mensch, Michaël Weiss

January 7, 2015

1 Introduction

We consider an adversarial multi-arm bandit problem with *side information*. Such a problem, located between *full-information* problem (at each round, the player observe the loss of every arm), and the *bandit problem* (at each round, the player observe only the loss for the arm he choose). It can model several real-life situations, an important one being preference observation on social networks : feeding a user with a given content, we can observe its feedback (*retweet* on Twitter, *like* on Facebook), along with the feedback of some of his friends virally fed with the same content.

The *bandit* problem can be addressed most efficiently using the well-know `Exp3` algorithm, described in [**?**]. Very recently, [**?**] has proposed to extend this algorithm to the *side-information problem*, where feedback distribution is modeled using Erdos-Renyi random graphs. Most importantly, this article focuses on ways to estimate loss distribution over arms, not knowing the underlying edge-revelation probability. It draws upper-bounds on maximum expected regret, using methods inspired from geometrical resampling, present in [**?**].

Erdos-Renyi graph are too simple to successfully model social networks, as population clusters leads to non-uniform connection probabilities. More complex models has been proposed for this purpose, including stochastic block models, which extends Erdos-Renyi to graphs with different clusters. We propose an algorithm that extends `DuplExp3` from [**?**] to stochastic block models, where bandit cluster label is known. We empirically show that our algorithm outperform [**?**] algorithm on any stochastic blockmodel graph with more than 2 clusters.

2 Exp3 on Erdös-Rényi Graph

We first recall [**?**] settings, and describe the algorithm adapted to Erdös-Renyi graphs, and its underlying principles. The problem is reduced to the situation where all non-chosen arms reveal their loss with unknown probability $r_{11} = r$. This can be modeled by Erdös-Rényi random graphs with parameter r: at each step t, for all (i, j) node pairs, we construct an edge between i and j with the probability r; we observe loss from the neighbors of chosen arm I_t.

Erdös-Rényi graph models interaction between a fully-connected group of people, where content is shared with probability r. Although it seems very simple, it already requires subtle adaptation of vanilla `Exp3` algorithm for the player to benefit from side-observation.

2.1 Problem Definition

We consider a sequential set on interactions with the multi-armed bandit we assume to have N arms, for each step $t = 1, .., T$ these are the actions performed by the environment :

1. The environment chooses losses for every arm noted $l_{t,i}$ for the arm i at the step t.

2. Following the algorithm we hope would minimize as much as possible the regret the player draws an arm I_t.

3. The player receives the loss l_{t,I_t}.

4. We define $(O_t)_{i\in[N]}$ as the indicative function of observed loss at step t. We have:

$$O_{t,I_t} = 1 \qquad \forall i \neq I_t,\, O_{t,i} \sim B\,(r\,)$$

   $(O_t)_{i\in[N]}$ corresponds to the value of the logic expression i *is neighbor of* I_t in the Erdös-Rényi random graph drawn at step t.

5. For all i such that $O_{t,i} = 1$ the player can observe the loss $l_{t,i}$.

We write $p_{t,i} = \mathbb{P}[I_t = i|\mathcal{F}_{t-1}]$ where $\mathcal{F}_{t-1}$ corresponds to all the actions and observations the player had until the step t. Then intuitively the probability of observing the loss of the arm i at the step t would be $q_{t,i} = p_{t,i} + (1 - p_{t,i})r$ and to use the Exp3 the the loss estimate :

$$\hat{l}_{t,i} = \frac{O_{t,i} l_{t,i}}{q_{t,i}}.$$

But the main problem resides in the fact that r is unknown so the algorithms presented use tricks to obtain loss estimates such that we keep the property :

$$\mathbb{E}[\hat{l}_{t,i}|\mathcal{F}_{t-1}] = l_{t,i}$$

The principal idea is to have access to two **independent** geometrically distributed random variables M_t^* and $K_{t,i}$ with respective parameters r and $p_{t,i}$, then the variable $G_{t,i}^* = \min\{K_{t,i}, M_t^*\}$ is also geometrically distributed with the parameter $q_{t,i}$ previously defined. Then if we have $G_{t,i}^*$ **independent** of $O_{t,i}$ we can replace in the definition of $\hat{l}_{t,i}$, $\frac{1}{q_{t,i}}$ by $G_{t,i}^*$.

2.2 DuplExp3 for large values of r

We assume $r \geq \frac{log(T)}{2N}$, which implies that the probability of having no additional observations in round t is bounded by $\frac{1}{\sqrt{T}}$.
This algorithm needs two Exp3 sub-algorithms, with learning rates (η_t). one for the round when t is even and the other one for the rest so that we can construct independent M_t^* and $K_{t,i}$ and independent $G_{t,i}^*$ and $O_{t,i}$. For each t, the algorithm draws:

$$p_{t+2,i} \propto w_{t+2,i} = \frac{1}{N} exp\left(-\eta_{t+2}\hat{L}_{t,i}\right)$$

Where $\hat{L}_{t,i} = \sum_{k=0}^{t/2} \hat{l}_{t-2k,i}$ the cumulative sum of the loss estimates for the arm i for one of the Exp3 sub-algorithms.

M_t^*, truncated geometrical variable of parameters r is constructed as such : For all $i = 1, ..., N-1$, we define $O'_{t,i}$ as:

$$\forall i < I_t\ O'_{t,i} = O_{t,i} \qquad \forall N \geq i > I_t\ O'_{t,i-1} = O_{t,i}$$
$$M_t^* = \min\{1 \leq i < N : O'_{t-1,i} = 1\} \cup \{N\}$$

We also define $K_{t,i}$ as a geometric random variable with parameter $p_{t,i}$ computed a the step $t-2$. The since M_t depends of O_{t-1} and $p_{t,i}$ of $(O_k)_{k\leq t-2}$ they are obviously independent. That's why we can consider :

$$G_{t,i} = \min\left(K_{t,i}, M_t\right)$$

with $G_{t,i}$ independent of $O_{t,i}$. $G_{t,i}$ follows a geometrical law of parameter $p_{t,i} + (1 - p_{t,i})r$.

We set the loss estimate as:

$$\hat{l}_{t,i} = G_{t,i} O_{t,i} l_{t,i}$$

which, taking the expectation, yields:

$$\begin{aligned}
\mathbb{E}\,\hat{l}_{t,i} &= \mathbb{E}\,G_{t,i}\,\mathbb{E}\,O_{t,i}\,\mathbb{E}\,l_{t,i} \\
&= \frac{1}{p_{t,i} + (1 - p_{t,i})r}\,(p_{t,i} + (1 - p_{t,i})r)\,\mathbb{E}\,l_{t,i} \\
&= \mathbb{E}\,l_{t,i}
\end{aligned}$$

which is unbiased estimator of $l_{t,i}$, as independence allows us to separate expectation. Setting η_t =, the following upper-bound on the regret can be drawn, using unbiased estimator $\hat{l}$:

$$R_T \leq 4\sqrt{\left(\frac{T}{r} + N^2\right)\log N} + \sqrt{T}$$

2.3 Lower-bounding r

However, a problem remains in this algorithm ; since we don't know a priori what is the value of r, we can't ensure that $r \geq \frac{log(T)}{2N}$ as did the assumption previously. So we need to find a lower bound on r to know in which case we probably are. The algorithm `Estimate` $\underline{r}$ returns the argument $\underline{r}$ with the following properties :

$$\begin{aligned}
&\mathbb{P}[\underline{r} \leq r] \geq 1 - \frac{1}{\sqrt{T}} \\
&\mathbb{P}[\underline{r} = 0] = 1 - \frac{1}{\sqrt{T}}, \quad if\ r \leq \frac{1}{N} \\
&\mathbb{P}[\underline{r} = 0] \leq \frac{1}{\sqrt{T}}, \quad if\ r \geq \frac{2}{N}
\end{aligned}$$

Furthermore, let τ be the index of the round when `Estimate` $\underline{r}$ terminates, then we have :

$$\begin{aligned}
&\mathbb{E}\,[\tau] = \frac{4\ log\ T}{N} + \sqrt{T} + 1,\ if\ r \leq \frac{1}{N} \\
&\mathbb{E}\,[\tau] = log\ T\left(\frac{4}{N} + \frac{e}{2}\right) + 2,\ if\ r \geq \frac{2}{N} \\
&\mathbb{E}\,[\tau] = log\ T\left(\frac{4}{N} + e\right) + 2,\ if\ r \in [1/N, 2/N]
\end{aligned}$$

These properties will allow us to run this algorithm in first before running a learning algorithm without increasing too much the expected regret. In this configuration, then we can safely estimate that if :

$$\underline{r} \geq \frac{log\ T}{N}$$

Then we can consider that r is probably large enough to run the previously algorithm DuplExp3 on the graph. We will now see how we should proceed in the other cases.

2.4 Generalized algorithm

We now are able to generate a safe lower bound of r, $\underline{r}$. This is the first step of the algorithm. We already explained that we could resume to DuplExp3 at the current step τ if we had $\underline{r} \geq \frac{log\ T}{N}$. We now explain what are the solutions in the other cases.

If $\underline{r} = 0$:
We consider that r is really equal to 0 and run vanilla Exp3 with parameter $\eta = \sqrt{(2\ log\ N)\,/\,(TN)}$ from the step τ.

If $0 < \underline{r} < \frac{log\ T}{N}$:
Then r might be too small so we can have have side information and so have a biased estimation r during the steps. In this case, we use a simple trick to reattach to the case of DuplExp3 : The new algorithm groups multiple steps together so that it was as if the number of episodes became,

$$J = \left\lceil \frac{T}{A} \right\rceil$$
$$where\ A = \left\lceil \frac{logT}{N\underline{r}} \right\rceil$$

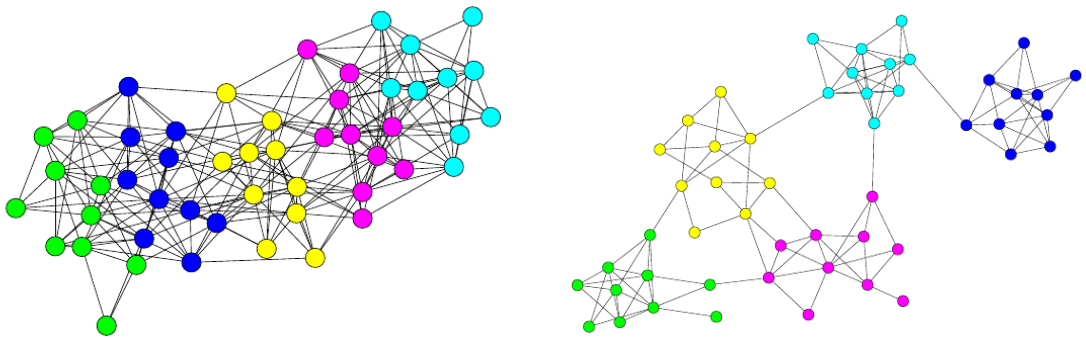
With some adaptations, for instance I_t should be the same for A real rounds and we make sure none of the O_{t,I_t} are counted in the new computation of M_j - where $A(j-1) < t \leq Aj$ which are the grouped steps - and leads less biased estimations of $\hat{l}_{j,i}$.

3 Exp3 on Stochastic Block-Model Graphs

We consider a multi-armed bandit where the arms are divided up in several classes. Every time the learner chooses an arm, not only does he observes the loss of this arm, but he can also have information about losses of non-chosen arms. We model observation graphs by stochastic block models. We consider that when the learner choose a arm of the class i, the other arms of the class i have the probability r_{ii} to reveal their loss and the other arms of a class j have the probability r_{ij}. The problem is characterized by the matrix R which represents the probabilities of communication between two arms of different classes. In this problem, we assume that the classification of the arm as already been done, so we know the class of each arm and we define N_i, the size of the cluster i, $i = 1, ..., n$ where n represents the number of clusters.

3.1 Problem definition

As in Sec. 1, everything happens as if, at every time step t, we built a stochastic block-model graph, knowing bandit labels. Such graph can model a variety of group interaction, from dissociation to entanglement and layer organisation. We present an example of stochastic block model in Fig 1, with 5 clusters, cluster i being closely connected to cluster $i-1$ and $i+1$ only on the left, and cluster being closely connected from within on the right.



(a) ordered communitites (b) Assortative communities

Figure 1: Stochastic block model random graph (from [**?**])

Notations and definition are mostly the same as in Sec. 1, and will be introduced as we adapt [**?**] algorithm.

3.2 Algorithm

[**?**] algorithm aims at computing an unbiased estimator for each ($l_{i,t}$, in order to yield satisfying upper bounds on maximum expected regret. We follow the same principle, making use of arms' cluster labels. If we had access to every values r_{ij}, we would be able to define, choosing arm I_t at round t :

$$\hat{l}^*_{t,i} = \frac{O_{t,i} l_{t,i}}{p_{t,i} + (1 - p_{t,i}) r_{I_t,i}}$$

Keeping with the original algorithm, we thus want to build an estimator $G_{i,t}$ that follows a geometrical law of parameter $p_{t,i} + (1 - p_{t,i}) r_{I_t,i}$. We will rely on the sampling of a geometric variable M_t of parameter $r_{I_t,i}$, making sure that it stays independent from $O_{t,i}$. That way, setting

$$\hat{l}_{t,i} = G_{t,i} O_{t,i} l_{t,i}$$

we obtain the desired unbiased estimator. The whole difficulty thus rely on the sampling of M_t so that it stays independent of O_t.

M_t sampling

3.3 Adaptation of Estimate r, generalized algorithm

We now want to provide lower-bounds for every r_{ij} so we adapt the previous algorithm for every r_{ij}, we compute C_i which depends of N_i the same way C depended of N in [**?**] for the algorithm `estimate r`. For the following explanations, we consider variables x_{ij} which are computed when we draw I_t with uniform probability in the cluster i and look at the results in the cluster j, i.e. we look at the $O_{t,k}$ where k is in the cluster j.
Then for each cluster i we compute c_{ij} associated for each $j = 1, ..., n$ allowing to safely estimate between 0 and $\max_i(C_i)$ if $\underline{r}_{ij} = 0 \ \forall j = 1, ..., n$. So this step costs $k * \max_i(C_i)$ iterations.
The second step needs also to estimate $\underline{r}_{ij}$ if it had not been estimated as equal to 0 during the first step. We also perform it linearly for each cluster i where we compute m_{ij} and $M_{ij}[m_{ij}],\ m = 1, .., k, j = 1, .., k$ in parallel, the equivalents of j and $M[j]$ in [**?**].
Finally when all end conditions are met we return $\underline{R}$. What is interesting about the algorithm is that it provides $\underline{R}$ such that :

$$\mathbb{E}\,\tau \leq O\left(k\sqrt{T}\right) \ \ if\ \forall i = 1, ..., n\ ,\ \exists j \in 1, ..., N \ and \ r_{ij} < \frac{1}{N_j}$$

in the worst case scenario. Besides we have the probabilities :

$$\begin{aligned}
\mathbb{P}[\underline{r}_{ij} \leq r_{ij}] &\geq 1 - \frac{1}{\sqrt{T}} \\
\mathbb{P}[\underline{r}_{ij} = 0] &= 1 - \frac{1}{\sqrt{T}}, \quad if\ r_{ij} \leq \frac{1}{N_j} \\
\mathbb{P}[\underline{r}_{ij} = 0] &\leq \frac{1}{\sqrt{T}}, \quad if\ r_{ij} \geq \frac{2}{N_{ij}}
\end{aligned}$$

So these lower bounds are still pretty safely estimated.

After computing $\underline{R}$, we can know if at a step t, after choosing I_t if we need to group so steps to have unbiased loss estimates. This could be done considering $\underline{r}_N^* = \min_{j,\ \underline{r}_{i,j}>0}\ \underline{r}_{i,j} N_j$ and then using to group the next rounds $A^* = \left\lceil \frac{\log T}{\underline{r}_N *} \right\rceil$, but would make depend A^* of I_t and so might change the number of steps we group at each round. To have a general A which would not depend of I_j we chose at round j we consider :

$$\underline{r}_N^* = \min_{i,j,\ \underline{r}_{i,j}>0} \underline{r}_{i,j}\ N_j$$

$$A = \left\lceil \frac{logT}{\underline{r}_N^*} \right\rceil$$

Then we can group A steps for each rounds and apply the previously described algorithm.

If $\underline{R} = 0$:

Then we consider as if we had no sides information, no matter in which cluster the arm I_t is picked therefore we can perform vanilla Exp3 and drop the information about clusters.