A Batched Successive-Elimination Policy

For completeness, we include a description of the BaSE algorithm from Gao et al. [2019] below. The algorithm itself is quite simple: it just eliminates all arms that are suboptimal with high probability after each batch.

Algorithm 3: Batched Successive Elimination (BaSE) from Gao et al. [2019]

Input: Number K of arms, time horizon T, number of batches B, grid $t_0 = 0 < t_1, \ldots, t_B = T$, parameter γ.
Set $A = [K]$ (this will be the set of alive arms).
For each $j \in [K]$, initialize $n_j = 0$ and $\sigma_j = 0$.
for $b \leftarrow 1$ **to** $B - 1$ **do**
  for $t \leftarrow t_{b-1} + 1$ **to** t_b **do**
    Let j be the $(t \bmod |A|)$th element of A.
    Play arm j and receive reward x_t.
    $n_j \leftarrow n_j + 1$. $\sigma_j \leftarrow \sigma_j + x_t$.
  end
  Let $j^* = \arg\max_{j \in A} \sigma_j / n_j$.
  Let $\tau = \max_{j \in A} n_j$ (by construction, all n_j for $j \in A$ should be approximately equal).
  for $j \in A$ **do**
    if $\sigma_{j^*}/n_{j^*} - \sigma_j/n_j \geq \sqrt{\gamma/\tau}$ **then**
      Remove j from A.
    end
  end
end
for $t \leftarrow t_{b-1} + 1$ **to** t_b **do**
  Pull arm $j^* = \arg\max_{j \in A} \sigma_j / n_j$.
end

The authors show that when $B = \log\log T$, $\gamma = O(\log(KT))$, and $t_b = O(T^{1-1/2^b})$ the above algorithm incurs at most $\tilde{O}(\sqrt{KT})$ regret. In our applications, we will set $\gamma = O(\log(NKT))$; this guarantees that the probability the best arm is ever eliminated from A is at most $1/\mathrm{poly}(N, K, T)$ (see Lemma 1 of Gao et al. [2019]), and therefore with high probability Algorithm 1 will never have to abort.

B Simulations

In this appendix we empirically evaluate the learning algorithms we introduce on this paper on several synthetic problem instances. We implement the following algorithms:

1. The Explore-Then-Commit algorithm (Algorithm 2) of Section 3.5.
2. Algorithm 1 with *greedy decomposition*, as defined in Section 3.4.1. Recall that this decomposition algorithm simply assigns each of the users to the same arm, cycling through the arms until all the demand is met.
3. Algorithm 1 with *random decomposition*, as defined in the beginning of Section 3.4.2. This decomposition algorithm assigns each user to a random one of their active arms, and generates such assignments until all demand is met.
4. Algorithm 1 with *LP-based decomposition*, as defined in Corollary 1 of Section 3.4.2. This decomposition algorithm writes the demand vector as a convex combination of the vertices of the polytope $\mathcal{P}_C([K])$ (each of which corresponds to a specific assignment).
5. A non-anonymous UCB algorithm, where each user independently runs UCB over the K arms.

We evaluate these algorithms on two classes of instances, both with $N = 50$ users, $K = 5$ arms, anonymity parameter $C = 4$, and $T = 10^5$ rounds. In the first class of instances (Figure 1), the

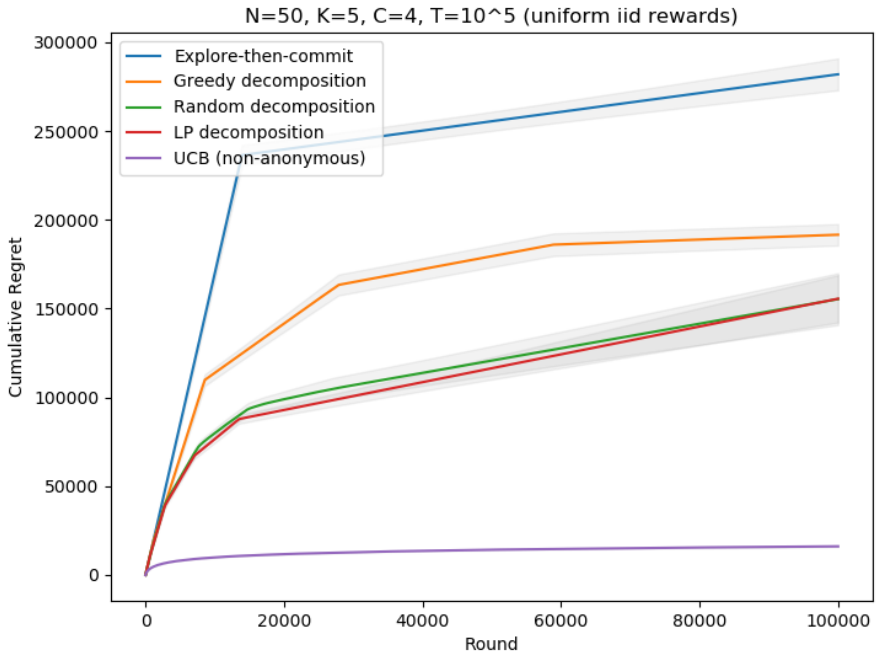

Figure 1: Cumulative regrets over time of five different learning algorithms on instances with uniform iid rewards (i.e., each user/arm distribution is a Bernoulli distribution with parameter independently drawn from $U([0,1])$). Grey regions represent 95% confidence intervals.

rewards for user i and arm j are drawn from a Bernoulli μ_{ij} distribution, where μ_{ij} is sampled from the uniform distribution $U([0,1])$ independently from all other means. The second class of instances (Figure 2) is generated by the following linear model: each user i is assigned a random unit norm 10-dimensional vector v_i and each action j is assigned a random unit norm 10-dimensional vector w_j. The rewards for user i and arm j are drawn from a Bernoulli μ_{ij} distribution, where $\mu_{ij} = 0.5(\theta_{ij} + 1)$, and where θ_{ij} is the cosine similarity between v_i and w_j.

With these parameters, both classes of instances almost always satisfy the user-cluster assumption for $U \geq C + 1$, so the preconditions to apply Algorithm 1 (and the various decomposition algorithms) almost always hold. Nonetheless, we implement Algorithm 1 semi-robustly: e.g., instead of aborting when we don't see a U-batched graph, we continue assigning users to random active arms. Similarly, we implement all algorithms so that they are agnostic to U (essentially, we set $U = C + 1$ in Algorithm 1 and compute an appropriate lower-bound on the approximation factor α). We optimize various hyperparameters of our algorithm (e.g. the learning rate of BaSE) by validation on an independent set of learning instances. See the attached code for additional details.

Figure 1 shows the average cumulative rewards (and confidence intervals) of twenty independent runs of these algorithms on the class of instances with uniformly generated rewards. Unsurprisingly, the one algorithm which violates anonymity (parallel UCB) does much better than all the C-anonymous algorithms, obtaining a total regret about 10 times smaller than the next better (this is consistent with our regret bounds, which indicate that at best our anonymous algorithms incur at least $\Omega(C)$ more regret than non-anonymous algorithms). The ordering of the various anonymous algorithms is also as expected. Explore-then-commit (which works without any guarantee on U and only uses one "batch") performs significantly worse than the variations of Algorithm 1 based off batched bandit algorithms. Within the variations of Algorithm 1, the naive greedy decomposition performs worst. The variations with random decomposition and LP decomposition perform similarly; this is not too surprising given that the LP decomposition is in some sense a derandomization of the random decomposition.

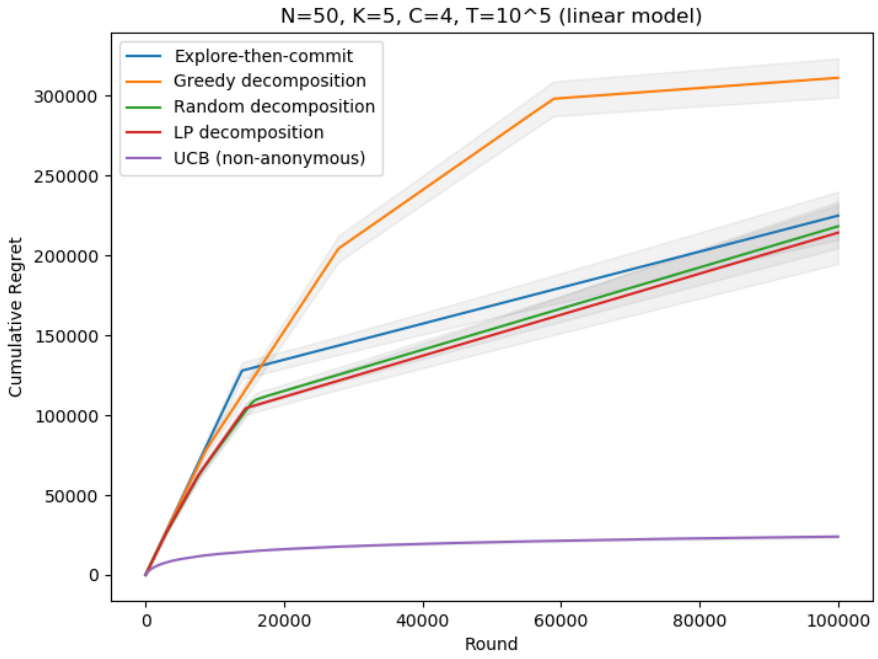

Figure 2: Same as Figure 1, but where reward distributions arise from a linear model instead of being generated uniformly (see text).

Figure 2 shows the average cumulative rewards (and confidence intervals) of twenty independent runs of these algorithms on instances generated by the linear model defined above. In most aspects, it is very similar to Figure 1; one major qualitative difference, however, is that greedy decomposition appears to perform much worse on this class of instances (whereas greedy decomposition outperformed explore-then-commit in Figure 1, it does markedly worse than explore-then-commit here).

C Omitted proofs

C.1 Proof of Lemma 1

Proof of Lemma 1. First note that with a simple Hoeffding bound for an i and a b we have

$$\begin{aligned}
&Pr\Big(|\bar{Y}^i(t_b) - \mu_i| \leq \frac{1}{2}\sqrt{\frac{\gamma \log(TK)}{\tau_b}}\Big) \\
&< 2\exp\Big(-2\tau_b \frac{1}{4}\frac{\gamma \log(TK)}{\tau_b}\Big) \\
&= 2\exp\Big(-0.5\gamma \log(TK)\Big) \\
&\leq \frac{1}{(TK)^2}. \qquad \text{for large enough constant } \gamma \text{ and } K \geq 2
\end{aligned}$$

By union bound this holds for all i and b with probability at least $1 - \frac{1}{TK}$. In the rest, w.l.g. we assume this holds, since $D_b \cdot \mathbb{E}\left[\max_{j \in A_b} \Delta_j\right] \times \frac{1}{TK} \leq \frac{D_b}{TK} \leq 1$. This means that for any j that

survives round $b-1$, (i.e. $\forall j \in A_b$) we have

$$\begin{aligned}\Delta_j = \mu^* - \mu_j &\leq 2\sqrt{\frac{\gamma \log(TK)}{\tau_{b-1}}} \\ &\leq 2\sqrt{\frac{\gamma \log(TK)}{\frac{u_{b-1}}{K}}} \\ &= 2\sqrt{\frac{\gamma K \log(TK)}{a^{2-2^{2-b}}}}\end{aligned}$$

Therefore we have

$$\begin{aligned}D_b \cdot \mathbb{E}\left[\max_{j \in A_b} \Delta_j\right] &\leq D_b \times 2\sqrt{\frac{\gamma K \log(TK)}{a^{2-2^{2-b}}}} \\ &\leq u_b \times 2\sqrt{\frac{\gamma K \log(TK)}{a^{2-2^{2-b}}}} \\ &= a^{2-2^{1-b}} \times 2\sqrt{\frac{\gamma K \log(TK)}{a^{2-2^{2-b}}}} \\ &= \frac{a^{2-2^{1-b}}}{a^{1-2^{1-b}}} \times 2\sqrt{\gamma K \log(TK)} \\ &= 2a\sqrt{\gamma K \log(TK)} \\ &= \Theta\Big(T^{\frac{1}{2-2^{1-B}}}\sqrt{\gamma K \log(TK)}\Big) \\ &= \Theta\big(\sqrt{\gamma TK \log(TK)}\big).\end{aligned}$$

□

C.2 Proof of Lemma 2

Proof of Lemma 2. Assume (as above) that user i is the kth user in G_s. Let $G'_s = G_{s,k} = G_s \setminus \{i\}$. For each user $i' \in G_s$, let $X_{i'}$ denote the r.v. representing the reward user i' contributed in round 0 of the above procedure, and for each user $i' \in G'_s$, let $Y_{i'}$ denote the r.v. representing the reward user j contributed in round k of the above procedure. Note that (by the definition of our setting), the value $X_{i'}$ and $Y_{i'}$ are independent r.v.s with $\mathbb{E}[X_{i'}] = \mathbb{E}[Y_{i'}] = \mu_{i',\pi(i')}$. In addition, $|X_{i'}|, |Y_{i'}| \in [0,1]$, so $\mathrm{Var}(X_{i'}), \mathrm{Var}(Y_{i'}) \leq \frac{1}{4}$.

Then, since

$$\hat{\mu}_{i,\pi(i)} = r_{s,0} - r_{s,k} = X_i + \sum_{i' \in G'} (X_{i'} - Y_{i'}),$$

it follows that

$$\mathbb{E}[\hat{\mu}_{i,\pi(i)}] = \mathbb{E}[X_i] + \sum_{i' \in G'_s} (\mathbb{E}[X_{i'}] - \mathbb{E}[Y_{i'}]) = \mathbb{E}[X_i] = \mu_{i,\pi(i)}.$$

Furthermore, note that each of the r.v.s $X_{i'}$ and $Y_{i'}$ are 1-subgaussian. Since $\hat{\mu}_{i,\pi(i)}$ is the sum of at most $4C+1$ independent 1-subgaussian variables, $\hat{\mu}_{i,\pi(i)}$ itself is $(4C+1)$-sub-Gaussian (and hence $O(C)$-subgaussian).

□

C.3 Proof of Theorem 1

Proof of Theorem 1. Fix a user i. We will show that the expected regret incurred by user i is at most $\tilde{O}(C\sqrt{\alpha KT})$, thus implying the theorem statement. As mentioned in Appendix A, the guarantees

of BaSE imply that with high probability this algorithm will never abort, so from now on we will condition on the algorithm not aborting.

Consider the expected regret incurred by user i during batch b of Algorithm 1. Since i is only ever assigned to arms in $A_{i,b}$ during this batch, and since user i gets matched a total of $\alpha(2C+2)D_b$ times during this batch, this expected regret is at most

$$\alpha(2C+2)D_b \cdot \mathbb{E}\left[\max_{j \in A_{i,b}} \Delta_{i,j}\right],$$

where $\Delta_{i,j} = (\max_{j'} \mu_{i,j'}) - \mu_j$. On the other hand, by Lemma 1 (and using the fact that the feedback we provide to BaSE is $O(C)$-subgaussian by Lemma 2), this is at most

$$\alpha(2C+2) \cdot \tilde{O}\left(\sqrt{C} \cdot \sqrt{KT'}\right) = \tilde{O}(C\sqrt{\alpha KT}),$$

as desired. □

C.4 Proof of Lemma 3

Proof of Lemma 3. By Caratheodory's theorem, if $\beta w \in \mathcal{P}_C$, we can write βw as the convex combination of at most $NK+1$ vertices $v^{(1)}, v^{(2)}, \dots v^{(NK+1)} \in \mathcal{P}_C$. Let us write

$$\beta w = \sum_{\ell=1}^{NK+1} \lambda_\ell v^{(\ell)} \tag{2}$$

where $\sum_\ell \lambda_\ell = 1$. In particular, we have that:

$$Dw = \sum_{\ell=1}^{NK+1} \frac{D\lambda_\ell}{\beta} v^{(\ell)}. \tag{3}$$

Consider the decomposition which contains $\lceil D\lambda_\ell/\beta \rceil$ instances of the assignment $v^{(\ell)}$ for each $1 \le \ell \le NK+1$ (here, "the assignment $v^{(\ell)}$" refers to any assignment which sends i to j if $v_{ij}^{(\ell)} = 1$). By (3), this assignment is guaranteed to contain at least $Dw_{ij} = D/|A_i|$ assignments which are informative for the pair (i,j), so this is a valid C-anonymous decomposition. Moreover, the number of assignments in this decomposition is at most

$$\sum_{\ell=1}^{NK+1} \left\lceil \frac{D\lambda_\ell}{\beta} \right\rceil \le (NK+1) + \sum_{\ell=1}^{NK+1} \frac{D\lambda_\ell}{\beta} = \frac{1}{\beta} D + NK + 1.$$

Likewise, if there are $R = \alpha D$ assignments $M_1, M_2, \dots M_R$ which form a C-anonymous decomposition of G, let $v^{(1)}, v^{(2)}, \dots, v^{(R)}$ be the vertices of $\mathcal{P}_C$ corresponding to these assignments. Then we are guaranteed that

$$Dw \le v^{(1)} + v^{(2)} + \cdots + v^{(R)},$$

so in particular we must have

$$\frac{1}{\alpha} w = \lambda \cdot \frac{v^{(1)} + v^{(2)} + \cdots + v^{(R)}}{R} + (1-\lambda) \cdot 0$$

for some $\lambda \in [0,1]$. Note that the RHS is a convex combination of vertices of $\mathcal{P}_C$ (in particular, $0 \in \mathcal{P}_C$), so $\frac{1}{\alpha} w \in \mathcal{P}_C$. □

C.5 Proof of Lemma 4

Proof of Lemma 4. Note that the vertices of $\mathcal{P}_C(S)$ satisfy all the constraints in (1), and every integral point satisfying (1) satisfies the conditions to be a vertex of $\mathcal{P}_C(S)$. It therefore suffices to show that the polytope defined by (1) is integral.

But this immediately follows, since the matrix of constraints in (1) are equivalent to the totally unimodular matrix defining the bipartite matching polytope between $[N]$ and $[K]$. □

C.6 Proof of Lemma 5

Proof of Lemma 5. We show w satisfies the constraints of (1) for $S = [K]$. The only nontrivial constraint is showing that

$$\sum_{i \in [N]} w_{ij} \geq C + 1, \ \forall j \in [K].$$

Note that by the definition of the user-cluster assumption, there are at least U choices of i for which $w_{ij} > 0$. By the definition of w_{ij}, if $w_{ij} > 0$, $W_{ij} \geq 1/K$. Since $U \geq K(C+1)$, the above inequality immediately follows. □

C.7 Proof of Theorem 2

Proof of Theorem 2. Note that, by construction, in the first T_{exp} rounds of this algorithm, we obtain $\approx T_{exp}/(K(2C+2))$ independent unbiased estimators $\hat{\mu_{i,j}}$ from the feedback-eliciting sub-algorithm for each user/arm pair (i, j). That is, for each (i, j) we are guaranteed that each $n_{ij} \geq T_{exp}/(K(2C+2))$ and that σ_{ij} is the sum of n_{ij} independent copies of an unbiased estimator for μ_{ij} with C-subgaussian noise.

It follows that $\sigma_{ij}/n_{ij} = m_{ij}$ is an unbiased estimator for μ_{ij} with C/n_{ij}-subgaussian noise. Let $\max_{ij} C/n_{ij} = \lambda$; note that $\lambda \leq KC(2C+2)/T_{exp} = \tilde{O}(K^{2/3}C^{4/3}T^{-2/3})$.

Since $m_{ij} - \mu_{ij}$ is λ-subgaussian, it follows from Hoeffding's inequality that

$$\Pr\left[|m_{ij} - \mu_{ij}| \geq \varepsilon\right] \leq 2\exp\left(-\frac{\varepsilon^2}{2\lambda}\right).$$

We will set $\varepsilon = \sqrt{4\lambda \log(NKT)}$; it then follows that

$$\Pr\left[|m_{ij} - \mu_{ij}| \geq \varepsilon\right] \leq \frac{2}{(NKT)^2}.$$

Union-bounding over all NK pairs $(i, j) \in [N] \times [K]$, with high probability (at least $1 - 1/T^2$) all estimates m_{ij} are within ε of μ_{ij}. Fix $i \in [N]$. It then follows that if we let $\hat{j} = \arg\max_j \sigma_{ij}/n_{ij}$, that $\mu_{i\hat{j}} \geq \max_j \mu_{ij} - 2\varepsilon$. In particular, for each remaining round after round T_{exp}, user i incurs at most ε regret.

The total expected regret from rounds up to T_{exp} is at most NT_{exp}. The total expected regret from rounds after T_{exp} is at most $NT\varepsilon$. It follows that the overall expected regret is at most $N(T_{exp} + T\varepsilon) = \tilde{O}(NC^{2/3}K^{1/3}T^{2/3})$ as desired. □

C.8 Proof of Corollary 1

Proof of Corollary 1. Follows from Lemmas 3 and 5. The decomposition of w into vertices of $\mathcal{P}_c([K])$ can be found efficiently from formulation (1) of $\mathcal{P}_c([K])$ as the intersection of a small number of half-spaces and following the algorithmic proof of Caratheodory's theorem (see e.g. Hoeksma et al. [2016] for details). □

C.9 Proof of Lemma 6

Proof of Lemma 6. Fix a, and set $w_{ij}^{(a)} = 1/|A_i \cap S_a|$ if $j \in A_i \cap S_a$, and $w_{ij}^{(a)} = 0$ otherwise. We first claim $w^{(a)} \in \mathcal{P}_c(S_a)$. As in Lemma 5, the only nontrivial condition to check is whether

$$\sum_{i \in [N]} w_{ij}^{(a)} \geq C + 1$$

holds for all $j \in S_a$. As before, for each $j \in S_a$, there must be at least U users i for which $j \in A_i$, and thus at least U users i where $w_{ij}^{(a)} > 0$. But now, if $w_{ij}^{(a)} > 0$, then $w_{ij}^{(a)} \geq 1/(U/(C+1)) = (C+1)/U$, and the above inequality follows.

To see that $w \leq \sum_{i=1}^{\alpha} w^{(a)}$, note that if $j \in A_i \cap S_a$, then $w_{ij} = 1/|A_i| \leq 1/|A_i \cap S_a| = w_{ij}^{(a)}$. $\square$

C.10 Proof of Corollary 2

Proof of Corollary 2. By Lemma 6, we can partition K into α sets S_a and write

$$\frac{1}{\alpha} w = \lambda \cdot \left(\frac{1}{\alpha} \sum_{a=1}^{\alpha} w^{(a)} \right) + (1 - \lambda) \cdot 0$$

where $\lambda \in [0, 1]$ and each $w^{(a)} \in \mathcal{P}_C(S_a)$. This proves that $\frac{1}{\alpha} w \in \mathcal{P}_C$, and thus such a decomposition exists by Lemma 3. Moreover, we can efficiently find such a decomposition by decomposing each of the terms $w^{(a)}$ into a convex combination of 0 (which lies in $\mathcal{P}_C(S)$ for all S) and at most $N|S_a|$ other vertices of $\mathcal{P}_C(S_a)$ (since $\mathcal{P}_C(S_a)$ is only $N|S_a|$-dimensional). $\square$

C.11 Proof of Theorem 3

Proof of Theorem 3. Consider the following variant of the anonymous bandits problem. As in the anonymous bandits problem, an instance of this problem is specified by a number of users N, a number of arms K, an anonymity parameter C, a time horizon T, and for every pair of user i and arm j a 1-subgaussian reward distribution $\mathcal{D}_{i,j}$ with mean $\mu_{i,j} \in [0, 1]$. However, unlike the anonymous bandits problem which has a centralized learner, in this variant each user is an independent learner – moreover, we prohibit users from communicating with one another or observing the feedback of other users (in this problem, users' actiuns will not interact). On the other hand, we will tell each user i all distributions $\mathcal{D}_{i',j}$ belonging to users $i' \neq i$ (so they only need to learn their own reward distributions).

During each round t each user i will choose both a target arm $j_{i,t}$ (as in anonymous bandits), *and also* a subset $F_{i,t} \subseteq [N] \setminus \{i\}$ of at least $C - 1$ other users. User i (indirectly) obtains reward $r_{i,j_{i,t}}$, but only observes as feedback the sum

$$r_{i,\pi_t(i)} + \sum_{i' \in F_{i,t}} r_{i',j_{i,t}},$$

where each $r_{i,j}$ is independently drawn from $\mathcal{D}_{i,j}$. The goal is to maximize the total reward among all users, and regret is defined analogously to how it is in the anonymous bandits problem.

We first claim the above problem is strictly easier than the anonymous bandits problem, in the sense that any algorithm for the anonymous bandits problem that achieves expected regret R on a specific instance can be converted to an algorithm for the above problem that achieves expected regret R on the analogous instance. To see this, fix an algorithm $\mathcal{A}$ for the anonymous bandits problem and a user i in our variant of the problem. Note that user i can accurately simulate $\mathcal{A}$ with their knowledge of other distributions $\mathcal{D}_{i',j}$ and the feedback provided in the variant: in particular, if $\mathcal{A}$ plays assignment π_t in round t, user i should set $j_t = \pi_t(i)$ and set $F_{i,t} = \pi_t^{-1}(i)$ (if $|\pi_t^{-1}(i)| < C$, $\mathcal{A}$ got no information on i in round t and the user can set $F_{i,t}$ arbitrarily). User i then receives as feedback the aggregate reward from their group in the current execution of $\mathcal{A}$, and can simulate

aggregate rewards from other groups by sampling from $\mathcal{D}_{i',j'}$. By doing this, user i receives the same expected reward in this problem as user i would in the analogous anonymous bandits problem.

We now show the above problem is hard. Consider the following distribution over instances of the above problem. Fix K and C, and choose an $N \gg K^2(C+1)$ and $T \gg \max(K, C, N)$. For each $i \in [N]$, choose an arm $j^*(i)$ uniformly at random from $[K]$ (this will be user i's unique favorite arm; since $N \gg K \cdot K(C+1)$, with high probability this assignment will satisfy the user-cluster assumption for $U = K(C+1)$). For each pair $i \in [N]$ and $j \in [K]$, let $\mathcal{D}_{i,j} = \mathcal{N}(0,1)$ if $j \neq j^*(i)$, and let $\mathcal{D}_{i,j} = \mathcal{N}\left(\sqrt{C/T}, 1\right)$ if $j = j^*(i)$.

Consider the problem faced by user i when faced with this distribution. If in round t user i selects arm j, then regardless of their choice of set $F_{i,t}$, they will observe a random variable drawn from $\mathcal{N}(\mu_{i,j} + M, C)$, where $M = \sum_{i' \in F_{i,t}} \mu_{i',j}$ is an offset term known to user i. After subtracting out M, this means that if user i selects arm j, they get an independent random variable from $\mathcal{N}(\mu_{i,j}, C)$. Since exactly one of the $\mu_{i,j}$ (as j ranges over $[K]$) equals $\sqrt{C/T}$ and all other $\mu_{i,j} = 0$, this is exactly the hard distribution for classic C-Gaussian bandits. It is known (see Chapter 15 of Lattimore and Szepesvári [2020]) that any algorithm must incur at least $\Omega(\sqrt{CKT})$ regret on this distribution of problem instances. It therefore follows that each user i incurs at least $\Omega(\sqrt{CKT})$, and therefore overall we incur at least $\Omega(N\sqrt{CKT})$ regret over all N users. □

C.12 Proof of Theorem 4

Proof of Theorem 4. For a fixed value of T, we will randomize uniformly between the following two instances. In both instances we will set $N = K = 3$, $C = 2$, and all reward distributions $\mathcal{D}_{i,j}$ will be Bernoulli distributions (defined by their mean $\mu_{i,j}$). Let $\varepsilon = T^{-1/3}$. In the first instance we set

$$\begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ \mu_{21} & \mu_{22} & \mu_{23} \\ \mu_{31} & \mu_{32} & \mu_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} - \varepsilon & \frac{1}{2} + \varepsilon \\ 0 & \frac{1}{2} + \varepsilon & \frac{1}{2} - \varepsilon \end{bmatrix},$$

and in the second instance we set

$$\begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ \mu_{21} & \mu_{22} & \mu_{23} \\ \mu_{31} & \mu_{32} & \mu_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} + \varepsilon & \frac{1}{2} - \varepsilon \\ 0 & \frac{1}{2} - \varepsilon & \frac{1}{2} + \varepsilon \end{bmatrix}.$$

Intuitively, user 1 likes only arm 1 (and arm 1 is only liked by user 1). User 2 either slightly prefers arm 2 to arm 3 or arm 3 to arm 2, and user 3 has the opposite preferences of user 2. We will let the random variable X denote which instance we are in, with $X = 1$ denoting the first instance and $X = -1$ denoting the second.

Consider the set of actions that reveal information about the value of X. Since $C = 2$, we must allocate at least 2 people to an arm. We know all the rewards (deterministically) for arm 1, so we must allocate this group of people to either arm 2 or 3. Finally, if we allocate both user 2 and user 3 to arm 2 or 3, we learn nothing about the instance we belong to (since they have symmetric actions). Therefore the only way to gain any information about X is to either allocate users $\{1, 2\}$ or $\{1, 3\}$ to one of arms 2 or 3.

Any of these allocations gives us equivalent information about X; in one of the two instances, we will receive a random variable $1 + \mathrm{Bern}(1/2 + \varepsilon)$, and in the other instance we will receive a random variable $1 + \mathrm{Bern}(1/2 - \varepsilon)$. In addition, in any of these allocations, we incur at least a net regret of 1 from querying this allocation (since we assign user 1 to an arm with reward 0). We call any assignment involving one of these allocations an "information-revealing assignment".

Assume we have an algorithm which sustains expected regret at most R. Since each information-revealing assignment incurs regret at least 1, by Markov's inequality the probability the algorithm performs more than $2R$ information-revealing assignments is at most $1/2$. Now, assume that at round t, the algorithm has performed only r information-revealing assignments. By the discussion above, this means that the only information the algorithm has regarding X is r samples from $\mathrm{Bern}(1/2 + X\varepsilon)$.

In general, the best statistical distinguisher between $\mathrm{Bern}(1/2+\varepsilon)$ and $\mathrm{Bern}(1/2-\varepsilon)$ requires at least C/ε^2 samples (for some constant $C > 0$) to distinguish these two distributions with probability at least $3/4$. Therefore, if $r \leq C/\varepsilon^2$, our learning algorithm cannot distinguish between $X = 0$ and $X = 1$ with probability greater than $3/4$, and is therefore guaranteed to incur at least $\Omega(\varepsilon)$ regret (by e.g. allocating user 2 to the wrong arm).

Now, if $2R < C/\varepsilon^2$, then in every round we have performed fewer than C/ε^2 information-revealing assignments, so we incur at least $\Omega(\varepsilon T) = \Omega(T^{2/3})$ regret. On the other hand, if $2R \geq C/\varepsilon^2$, then $R \geq C/2\varepsilon^2 = \Omega(T^{2/3})$, so in this case the algorithm also incurs at least $\Omega(T^{2/3})$ regret. □

C.13 Proof of Theorem 5

Proof of Theorem 5. For a fixed value of T, we will randomize uniformly between the following two instances. In both instances $N = K = C = 2$, and in both instances all reward distributions are completely deterministic. In the first instance we set $(\mu_{11}, \mu_{12}, \mu_{21}, \mu_{22}) = (1, 0, 0, 1)$ (user 1 likes arm 1 and user 2 likes arm 2); in the second instance we set $(\mu_{11}, \mu_{12}, \mu_{21}, \mu_{22}) = (0, 1, 1, 0)$ (user 1 likes arm 2 and user 2 likes arm 1).

Since $C = 2$, the only actions we can take which result in feedback are assigning both users to the same arm, but in both problem instances this results in an aggregate reward of 1 (and hence provides no information as to which instance we have chosen). By the choice of rewards, if an assignment receives reward x in the first instance, it receives reward $2 - x$ in the second instance – since it is impossible to learn any information about the choice of instance, this means any algorithm receives expected reward 1 per round. On the other hand, the optimal algorithm for each instance receives an expected reward of 2 per round. This implies the expected regret is at least T, as desired. □