k-experts - Online Policies and Fundamental Limits

Samrat Mukhopadhyay
Indian Institute of Technology (ISM)
Dhanbad

Sourav Sahoo
Indian Institute of Technology
Madras

Abhishek Sinha
Indian Institute of Technology
Madras

Abstract

We introduce the k-experts problem - a generalization of the classic *Prediction with Expert's Advice* framework. Unlike the classic version, where the learner selects exactly one expert from a pool of N experts at each round, in this problem, the learner can select a subset of k experts at each round ($1 \leq k \leq N$). The reward obtained by the learner at each round is assumed to be a function of the k selected experts. The primary objective is to design an online learning policy with a small regret. In this pursuit, we propose SAGE (**Sa**mpled Hed**ge**) - a framework for designing efficient online learning policies by leveraging statistical sampling techniques. For a wide class of reward functions, we show that SAGE either achieves the first sublinear regret guarantee or improves upon the existing ones. Furthermore, going beyond the notion of regret, we fully characterize the mistake bounds achievable by online learning policies for stable loss functions. We conclude the paper by establishing a tight regret lower bound for a variant of the k-experts problem and carrying out experiments with standard datasets.

1 INTRODUCTION

The classic *Prediction with Expert's Advice* problem, also known as the Experts problem in the literature, is a canonical framework for online learning (Cesa-Bianchi and Lugosi, 2006). This problem is usually formulated as a two-player sequential game played between a learner and an adversary. Consider a set of N experts indexed by the set $[N] = \{1, 2, \ldots, N\}$. At each round t, the adversary secretly selects a reward vector $\boldsymbol{r}_t \in [0,1]^N$ for the experts. At the same time (without knowing the rewards for the present round), the learner selects an expert (possibly randomly) and then receives a reward equal to the reward of the chosen expert. The goal of the learner is to design an online learning policy that incurs a small *regret*. Recall that the regret of an online learning policy over a given time horizon is defined as the difference between the reward accumulated by the best fixed expert in hindsight and the total expected reward accrued by the policy (see Eqn. (1)). Many online learning policies achieving sublinear regrets in this setting are known, most notably, Hedge (Vovk, 1998; Freund and Schapire, 1997).

In this paper, we initiate the study of the k-experts problem - a generalization of the above Experts framework. The k-experts problem arises in many settings, including online ad placement, personalized news recommendation, adaptive feature selection, and paging. In the k-experts problem, instead of selecting only one expert at each round, the learner selects a subset $S_t \subseteq [N]$ containing k experts at each round t ($1 \leq k \leq N$). The reward $q(S_t)$ received by the learner at round t depends on the rewards of the experts in the chosen set S_t. Table 1 lists some variants of the k-experts problem considered in this paper. In the Sum-reward variant, the reward accrued by the learner at round t is given by the sum of the rewards of the experts in the chosen set S_t. In particular, let $\boldsymbol{p}_{ti}$ denote the (conditional) marginal probability that the i^{th} expert is included in the set S_t, given the history $\mathcal{F}_{t-1}$ of the game up to round $t-1$. Then, we can express the (conditional) expected reward for the t^{th} round as $\mathbb{E}[q_{\texttt{sum}}(S_t)|\mathcal{F}_{t-1}] = \mathbb{E}[\sum_{i \in S_t} r_{ti}|\mathcal{F}_{t-1}] = \langle \boldsymbol{r}_t, \boldsymbol{p}_t \rangle$. However, unlike the Sum-reward variant, the expected accrued reward for other variants depends on higher-order joint inclusion probabilities as well (as opposed to only marginals). In our most general case, apart from monotonicity, we *do not* impose any other condition (*e.g.,* submodularity (Streeter and Golovin, 2007)) on the reward function. For each of the above variants, we consider the problem of designing an online expert selection policy that minimizes the regret $\mathcal{R}_T$ (or a

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

Table 1: Variants of the k-experts problem

Sum-reward	Max-reward	Pairwise-reward	Monotone reward
$q_{\texttt{sum}}(S_t) = \sum_{i \in S_t} r_{ti}$	$q_{\texttt{max}}(S_t) = \max_{i \in S_t} r_{ti}$	$q_{\texttt{pair}}(S_t) = \sum_{i,j \in S_t} r_{it} r_{jt}$	$q_{\texttt{monotone}}(S_t) = f_t(S)$

variant of it) over a horizon of length T:

$$\mathcal{R}_T = \max_{S:|S|=k} \sum_{t=1}^{T} q(S) - \sum_{t=1}^{T} \mathbb{E} q(S_t). \tag{1}$$

In the above, the expectation in the second term is taken with respect to any randomness of the learner.

Related work: A special case of the k-experts problem is *Online N-ary prediction with k-sets*, which we briefly refer to as the k-sets problem (Koolen et al., 2010). In this problem, a learner sequentially predicts the next symbol for an unknown N-ary sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$ chosen by an adversary. The symbols are revealed to the learner sequentially in an online fashion. However, instead of predicting a single symbol $\hat{y}_t \in [N]$ at each round, the learner is allowed to output a subset S_t, consisting of k symbols at round t. The learner's prediction for round t is considered to be correct if and only if the predicted set S_t contains the true symbol y_t. In the event of a correct prediction, the learner receives unit reward, else, it receives zero rewards for that round. The goal of the learner is to maximize its cumulative reward over a given time horizon. It is easy to see that the above problem is a special case of the k-experts problem with the Sum-reward variant, where the adversary's actions are constrained as $r_{ti} \in \{0, 1\}$ with $\sum_{i=1}^{N} r_{ti} = 1, \forall t, i$.

In a seminal paper, Cover (1966) studied the fundamental limits of online binary prediction, which is a special case of the k-sets problem with $N = 2$ and $k = 1$. Cover gave a complete characterization of the set of all *stable* reward profiles achievable by online policies (see Section 3 for the definition of stability). Fifty years later, Rakhlin and Sridharan (2016) generalized Cover's result to an arbitrary alphabet of size N, but still requiring $k = 1$. The characterization of the prediction error for the k-sets problem for an arbitrary N and k has been a long-standing open problem.

Coming back to the problem of minimizing the static regret for the k-sets problem, a quick-and-dirty approach can be used to reduce the problem to an instance of the classic Experts problem with a much larger set of experts, which we call *meta-experts*. In this reduction, a meta-expert is identified with one of the $\binom{N}{k}$ possible subsets of experts of size k. One can then use any known low-regret prediction policy, such as Hedge, on the meta-experts to design an online learning policy for the k-sets problem. Koolen et al. (2010) referred to the resulting Hedge policy as Expanded Hedge. An obvious challenge with this approach is to overcome the severe computational inefficiency of the resulting online policy, which, *apparently*, needs to keep track of exponentially many experts. To resolve this issue, Koolen et al. (2010) proposed the *Component Hedge* (CH) algorithm and showed that the proposed policy yields a tight regret bound. However, the CH algorithm involves a projection and decomposition step, each of which costs $O(N^2)$. Although the projection step was later shown to be implementable in linear time (Herbster and Warmuth, 2001, Theorem 7), the best-known algorithm for the decomposition step still takes $O(N^2)$ time (Warmuth and Kuzmin, 2008, Algorithm 2). Suehiro et al. (2012) speculate the existence of an $O(N \log N)$ algorithm for the decomposition. However, their algorithm (Algorithm 4) and its analysis mentioned in Theorem 10 of the paper still has $O(N^2)$ complexity. We refer the readers to Takimoto and Hatano (2013) for an excellent survey of the efficient projection and decomposition schemes for the k-sets and other online combinatorial optimization problems. The k-sets problem has also been investigated by Daniely and Mansour (2019), as an instance of the *paging* problem. The authors alleviated the complexity of the naive Hedge implementation by reducing it to a problem of sequential sampling from a recursively defined distribution. Unfortunately, the resulting policy is still sufficiently complex ($\Omega(N^2)$). Recently, Bhattacharjee et al. (2020) studied the paging problem and proposed an efficient and regret-optimal *Follow-the-Perturbed-Leader*-style policy. Although simple to implement, their algorithm does not admit an adaptive regret bound. Finally, Krause and Golovin (2014); Streeter and Golovin (2007); Harvey et al. (2020) studied online maximization of monotone submodular reward functions. However, the problem of achieving sublinear regret for arbitrary monotone reward functions has been wide open.

Our contributions: We make the following contributions in this paper:

1. In Section 1.1, we introduce SAGE - an efficient, projection-free, regret-optimal online prediction framework.
2. In Section 3, we generalize Cover (1966)'s result on binary sequence prediction by characterizing the set of all stable error profiles achievable by

online learning policies for the N-ary prediction problem with k-sets.

3. In Section 4, we design two online policies for the k-`sets` problem using the `SAGE` framework. The policy in Section 4.1 runs in linear time, admits an adaptive regret bound, and overcomes the existing quadratic computational barrier (see Table 2). The policy uses standard sub-routines such as Fast Fourier Transform and Madow's sampling. In Section 4.2, we propose another prediction policy for the k-`sets` problem based on the `FTRL` framework.

4. In Section 5, using the `SAGE` framework, we design an *improper* learning policy that achieves $O(\sqrt{T})$ regret for the `Pairwise-reward` variant of the k-`experts` problem.

5. In Section 6, we use the `SAGE` framework to design an efficient online prediction policy for arbitrary `Monotone reward` functions. This policy works by approximating the reward with modular functions. To the best of our knowledge, this is the *first* online learning policy for arbitrary monotone reward functions with a guaranteed $O(\sqrt{T})$ regret.

6. In Section 7, we establish a tight regret lower bound for the `Max-reward` version of the k-`experts` problem.

We conclude this section by giving a brief overview and key intuition for the `SAGE` framework.

1.1 Key Insights for `SAGE`

We begin our discussion with the `Sum-reward` variant in the k-`experts` problem. As pointed out earlier, the expected sum reward obtained by any policy depends *only* on the first-order marginal inclusion probabilities and *not* on the higher-order joint distribution. In particular, any two online prediction policies, that have the same conditional marginal inclusion probabilities, yield *exactly* the same reward per round. This simple observation leads to the `SAGE` meta-algorithm described in Algorithm 1. From the pseudocode, it is clear that the `SAGE` meta-algorithm has the same regret as the base policy π_{base}. However, unlike the base policy (which could be computationally intractable), the `SAGE` policy can be efficiently implemented in many problems. For example, we show in Section 4.1 that when `Hedge` is used as the base policy for the k-`sets` problem, the marginalization in line 3 reduces to the evaluation of certain elementary symmetric polynomials. These quantities can be efficiently computed using Fast Fourier Transform techniques. Furthermore, an efficient sampler for line 4 can be borrowed from the statistical sampling literature, reviewed in Section 2.

Algorithm 1 The Generic `SAGE` Meta-Algorithm

1: Start with a low-regret base online prediction policy π_{base} (*e.g.,* `Hedge`). We **do not** require the base policy π_{base} to be computationally efficient.
2: **for** each round t **do**
3: Efficiently compute the first-order marginal inclusion probabilities ($\boldsymbol{p}_t$) corresponding to the policy π_{base}. This step amounts to marginalizing the joint distribution induced by the policy π_{base}.
4: Efficiently sample k elements according to the marginal distribution $\boldsymbol{p}_t$ computed above.
5: **end for**

Note that `SAGE` is not necessarily regret-optimal for arbitrary monotone reward functions where the expected reward depends on higher-order inclusion probabilities. However, in Section 6, we show that we can still use the `SAGE` framework in this case by approximating the given reward function with modular reward functions. The approximation utilizes recent results from non-submodular set function optimization theory.

2 PRELIMINARIES: SAMPLING WITHOUT REPLACEMENT

The proposed `SAGE` meta-algorithm makes critical use of certain systematic sampling techniques from statistics (viz. line 4 of Algorithm 1). Consider the problem of sampling without replacement where one needs to randomly sample a k-set S from the universe $[N]$ such that item $i \in [N]$ is included in the set S with a prespecified marginal inclusion probability p_i, $\forall i \in [N]$. Formally, if the k-set S is sampled with probability $\mathbb{P}(S)$, we require that $\sum_{S: i \in S, |S|=k} \mathbb{P}(S) = p_i, \forall i \in [N]$. Since the sampling is done without replacement, for any k-set S, we have: $\sum_{i \in [N]} \mathbb{1}(i \in S) = k$. Taking expectation of both sides with respect to the randomness of the sampler, we conclude that any feasible marginal inclusion probability vector $\boldsymbol{p}$ must belong to the set Δ_N^k defined as follows:

$$\sum_{i \in [N]} p_i = k, \quad \text{and} \quad 0 \leq p_i \leq 1, \forall i \in [N]. \tag{2}$$

It turns out that condition (2) is also *sufficient* for designing efficient sampling schemes that leads to the marginal inclusion probability vector $\boldsymbol{p}$. Such sampling schemes have been extensively studied in the statistical sampling literature under the heading of *unequal probability sampling design* (Tillé, 1996; Hartley, 1966; Hanif and Brewer, 1980). In this paper, we use a linear-time exact sampling scheme proposed by Madow et al. (1949) as outlined below in Algorithm 2.

Table 2: Performance comparison among different policies for the k-sets problem

Policies	Reference	Regret bound	Complexity
FTPL (Gaussian perturbation)	Cohen and Hazan (2015)	$2\sqrt{2k^2T\ln\frac{Ne}{k}}$	$\tilde{O}(N)$
Component Hedge	Koolen et al. (2010)	$\sqrt{2kT\ln\frac{N}{k}}$	$O(N^2)$
SAGE (with $\pi_{\text{base}}=$ Hedge)	This paper	$\sqrt{2kT\ln\frac{Ne}{k}}$	$\tilde{O}(N)$
SAGE (with $\pi_{\text{base}}=$ FTRL)	This paper	$2\sqrt{2kT\ln\frac{N}{k}}$	$\tilde{O}(N)$

Algorithm 2 Madow's Sampling Scheme

Input: A universe $[N]$ of size N, cardinality of the sampled set k, and a marginal inclusion probability vector $\boldsymbol{p}=(p_1,p_2,\ldots,p_N)$ satisfying condition (2)
Output: A random k-set S with $|S|=k$ such that, $\mathbb{P}(i\in S)=p_i,\forall i\in[N]$
1: Define $\Pi_0=0$, and $\Pi_i=\Pi_{i-1}+p_i,\forall 1\le i\le N$.
2: Sample a uniformly distributed random variable U from the interval $[0,1]$.
3: $S\leftarrow\emptyset$
4: **for** $i\leftarrow 0$ to $k-1$ **do**
5: Select the element j if $\Pi_{j-1}\le U+i<\Pi_j$.
6: $S\leftarrow S\cup\{j\}$.
7: **end for**
8: **return** S

Correctness: The correctness of Madow's sampling scheme is easy to establish. From the necessary condition (2), it follows that Algorithm 2 selects exactly k elements. Furthermore, the element j is selected if the random variable $U\in\sqcup_{i=1}^{N}[\Pi_{j-1}-i,\Pi_j-i)$. Since U is uniformly distributed in $[0,1]$, the probability that the element j is selected is equal to $\Pi_j-\Pi_{j-1}=p_j,\forall j\in[N]$.

3 FUNDAMENTAL LIMITS OF ONLINE PREDICTION WITH k-sets

Consider the canonical binary prediction problem studied by Cover (1966). Assume that an adversary secretly selects a binary sequence $\boldsymbol{y}=(y_1,y_2,\ldots,y_T)$. The sequence is revealed to the learner one symbol at a time according to the following protocol - upon seeing the initial segment of the sequence $y_1^{t-1}\equiv(y_1,y_2,\ldots,y_{t-1})$ at time t, the learner makes a (randomized) guess $\hat{y}_t$ for the t^{th} element of the sequence y_t. The actual value of y_t is then revealed to the learner after the prediction. Let $\mu_{\mathcal{A}}(\boldsymbol{y})$ denote the fraction of mistakes made by a randomized prediction algorithm $\mathcal{A}$ for the sequence $\boldsymbol{y}$, *i.e.,* $\mu_{\mathcal{A}}(\boldsymbol{y})=\mathbb{E}^{\mathcal{A}}[T^{-1}\sum_{t=1}^{T}\mathbb{1}(y_t\neq\hat{y}_t)]$, where the expectation is taken with respect to the randomness of the prediction algorithm. In Eqn. (5) below, we show that irrespective of the prediction algorithm $\mathcal{A}$, the average fraction of errors $\mu_{\mathcal{A}}(\cdot)$ over all possible 2^T binary sequences is precisely $1/2$. A loss function $\phi:\{\pm1\}^T\to[0,1]$ is said to be *achievable* if there exists an online prediction policy $\mathcal{A}$ such that the average prediction error under the policy $\mathcal{A}$ for any sequence is upper bounded by the function ϕ, *i.e.,* $\mu_{\mathcal{A}}(\boldsymbol{y})\le\phi(\boldsymbol{y}),\forall\boldsymbol{y}$. An immediate question is to characterize the set of all achievable loss functions $\phi(\cdot)$.

For a given sequence $\boldsymbol{y}$, let $\phi(\ldots,j,\ldots)$ be a shorthand for the quantity $\phi(y_1,y_2,\ldots,y_{t-1},j,y_{t+1},\ldots,y_T)$. We call a loss function $\phi:\{\pm1\}^T\to[0,1]$ to be *stable* if it satisfies the following inequality for all $\boldsymbol{y}\in\{\pm1\}^T$ and for all time index $1\le t\le T$:

$$|\phi(\ldots,\underbrace{+1}_{t^{\text{th}}\text{ coordinate}},\ldots)-\phi(\ldots,\underbrace{-1}_{t^{\text{th}}\text{ coordinate}},\ldots)|\le\frac{1}{T}. \quad (3)$$

In this setup, Cover (1966) proved the following result:

Theorem 1 (Cover'66) *Suppose the loss function $\phi:\{\pm1\}^T\to[0,1]$ is stable. Then $\phi(\cdot)$ is achievable if and only if $\mathbb{E}\phi(\boldsymbol{z})\ge 1/2$, where the expectation is taken with respect to the i.i.d. uniform distribution over $\{\pm1\}^T$.*

We emphasize that although the statement of Theorem 1 involves an expectation, *no* probabilistic assumption was made on the sequence $\boldsymbol{y}$. Rakhlin and Sridharan (2016) extended Theorem 1 to the N-ary setting. In this paper, we generalize the result further to the k-sets setting where, instead of predicting a single value $\hat{y}_t$, the learner is allowed to predict a (randomized) subset $S_t\subseteq[N]$ containing k elements. Thus, the average loss incurred by a prediction policy $\mathcal{A}$ for the sequence $\boldsymbol{y}$ is given by:

$$\mu_{\mathcal{A}}(\boldsymbol{y})=\mathbb{E}^{\mathcal{A}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}(y_t\notin S_t)\right], \quad (4)$$

where the expectation is taken with respect to the randomness of the policy $\mathcal{A}$. Uniformly averaging the loss function $\mu_{\mathcal{A}}(\cdot)$ over all N^T possible N-ary sequences $\boldsymbol{y}$

(equivalently, endowing the set of all sequences in $[N]^T$ the i.i.d. uniform probability measure), we have

$$\begin{aligned}\mathbb{E}\mu_{\mathcal{A}}(\boldsymbol{y}) &= \mathbb{E}\mathbb{E}^{\mathcal{A}}\big[\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}(y_t \notin S_t)\big] \\ &\overset{\text{(Fubini's Th.)}}{=} \mathbb{E}^{\mathcal{A}}\mathbb{E}\big[\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}(y_t \notin S_t)\big] \\ &\overset{(a)}{=} 1-\frac{k}{N}, \end{aligned} \tag{5}$$

where (a) follows from the fact that $|S_t| = k, \forall t$. As in condition (3), we call a loss function $\phi : [N]^T \to [0,1]$ to be *stable* if for all sequences $\boldsymbol{y} \in [N]^T$ and all coordinates t of ϕ the following two conditions hold:

$$\max_{i\in[N]} \phi(\ldots, i, \ldots) - \frac{1}{N}\sum_{j\in[N]}\phi(\ldots, j, \ldots) \leq \frac{k}{NT}, \tag{6}$$

$$\frac{1}{N}\sum_{j\in[N]}\phi(\ldots, j, \ldots) - \min_{i\in[N]}\phi(\ldots, i, \ldots) \leq \big(1-\frac{k}{N}\big)\frac{1}{T}. \tag{7}$$

Our first result generalizes Cover's theorem by showing that conditions (6) and (7) together are also sufficient for the achievability.

Theorem 2 *Suppose the loss function $\phi : [N]^T \to [0,1]$ is stable. Then $\phi(\cdot)$ is achievable by some online policy if and only if $\mathbb{E}\phi(\boldsymbol{z}) \geq 1 - k/N$, where the expectation is taken w.r.t. the i.i.d. uniform distribution over $[N]^T$.*

The necessity part of Theorem 2 has already been established in Eqn. (5) above. The proof of sufficiency is constructive and proceeds in two phases. In Phase-I, at each round t, we compute a vector $\boldsymbol{p}_t$ satisfying the feasibility condition (2), such that p_{ti} gives the correct marginal inclusion probability of the element $i \in [N]$ that achieves the loss function $\phi(\cdot)$. In Phase-II, we sample a k-set $S_t \subseteq [N]$ according to the marginal inclusion probabilities $\boldsymbol{p}_t$ using Algorithm 2. Please refer to Section 11.1 of the supplementary material for the proof of Theorem 2.

Discussion: It is to be noted that directly using the generic online policy appearing in the achievability proof of Theorem 2 could be intractable in terms of computation or memory requirements. A more serious issue with the generic prediction policy is that it requires the loss function to be *stable*, which limits its applicability. Similar to the treatment in Rakhlin and Sridharan (2016), it might be possible to work with some relaxation of the loss function to derive a tractable policy. In the rest of the paper, we show that near-optimal inclusion probabilities may be efficiently computed via alternative methods, which result in low-regret efficient online prediction policies.

4 LEARNING POLICIES FOR THE k-sets PROBLEM

In this section, we propose two different efficient online policies for the k-sets problem. The first policy uses Hedge as the base policy and the second policy utilizes the standard *Follow-the-Regularized-Leader* framework.

4.1 k-sets with Hedge

For the simplicity of exposition, we use the the standard Hedge policy as our base policy in conjunction with the SAGE meta-algorithm. It will be clear from the sequel that any other Experts policy, such as Squint (Koolen and Van Erven, 2015) or AdaHedge (Erven et al., 2011), may also be used as the base policy, leading to more refined regret bounds.

1. The Base Policy: We start with the standard meta-experts framework as discussed in Section 1. Define a collection of $\binom{N}{k}$ experts, each corresponding to a distinct k-subset of the set $[N]$. Assume that the learner predicts the set S with probability $p_t(S), \forall S \in \binom{[N]}{k}$. The expected reward accrued by the learner when the adversary chooses symbol y_t at time t is given by:

$$\begin{aligned}&\mathbb{E}\big[\sum_{S:y_t\in S} 1\times\mathbb{1}(S_t = S) + \sum_{S:y_t\notin S} 0\times\mathbb{1}(S_t = S)\big] \\ &= \mathbb{P}(y_t \in S_t) = p_t(y_t), \end{aligned} \tag{8}$$

where $p_t(i) := \sum_{S:i\in S} p_t(S)$ is the marginal inclusion probability of the i^{th} element in the predicted k-set S. We now use the Hedge policy as our base policy for the resulting Experts problem. Let the indicator variables $r_\tau(i) := \mathbb{1}(y_\tau = i), \forall i$ encode the symbol chosen by the adversary at round τ. Furthermore, let the variable $r_\tau(S) := \sum_{i\in S} r_\tau(i)$ denote the reward accrued by the expert S at round τ. The cumulative reward accumulated by the expert S up to the round $t-1$ is given by $R_{t-1}(S) = \sum_{\tau=1}^{t-1} r_\tau(S)$. Overloading the notations a bit, let the variable $R_{t-1}(i)$ denote the number of times the i^{th} element appears in the sub-sequence $\boldsymbol{y}_1^{t-1}$. The Hedge policy with learning rate $\eta > 0$ chooses the expert S at round t with the following probability (Freund and Schapire, 1997; Vovk, 1998):

$$p_t(S) = \frac{w_{t-1}(S)}{\sum_{S'\subseteq[N]:|S'|=k} w_{t-1}(S')}, \quad \forall S \in \binom{[N]}{k}, \tag{9}$$

where $w_\tau(S) := \exp(\eta R_\tau(S))$.

2. Efficient Computation of the Inclusion Probabilities: The marginal inclusion probabilities for

each of the N elements can be obtained by marginalizing the joint distribution given by Eqn. (9). Let $w_{t-1}(i) := \exp(\eta R_{t-1}(i))$. We have

$$p_t(i) = \sum_{S:|S|=k, i\in S} p_t(S) = \frac{w_{t-1}(i)\sum_{S\subseteq[N]\setminus\{i\}:|S|=k-1} w_{t-1}(S)}{\sum_{S'\subseteq[N]:|S'|=k} w_{t-1}(S')}, \quad (10)$$

where we have used the fact that for any $S \subseteq [N]\setminus\{i\}$, we have $w_{t-1}(i)w_{t-1}(S) = w_{t-1}(S\cup\{i\})$. Clearly,

$$\sum_{i\in[N]} p_t(i) = \frac{\sum_{i\in[N]} w_{t-1}(i)\sum_{S\subset[N]\setminus\{i\}:|S|=k-1} w_{t-1}(S)}{\sum_{S'\subset[N]:|S'|=k} w_{t-1}(S')} \overset{(a)}{=} k, \quad (11)$$

where step (a) follows from the fact that for any k-set S, the term $w_{t-1}(S)$ appears in the numerator exactly k times. Therefore, the marginal inclusion probabilities in Eqn. (10) satisfy the feasibility condition (2). Hence, given the marginal inclusion probabilities, Algorithm 2 may be used to efficiently sample the predicted k-set. However, naively computing the marginal inclusion probabilities using Eqn. (10) requires evaluating sums of $\binom{N-1}{k-1}$ terms, which is computationally intractable. This difficulty can be alleviated upon realizing that both the numerator and denominator of Eqn. (10) can be expressed in terms of elementary symmetric polynomials as shown below. For any vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_N) \in \mathbb{R}^N$, define the associated *elementary symmetric polynomial* (ESP) of order l as:

$$e_l(\boldsymbol{w}) = \sum_{I\subseteq[N],|I|=l} \prod_{j\in I} w_j. \quad (12)$$

Furthermore, for any index $i \in [N]$, let $\boldsymbol{w}_{-i} \equiv (w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_N) \in \mathbb{R}^{N-1}$ denote the subvector with its i^{th} component removed. Then, from Eqn. (10), it follows that $p_t(i) = \frac{w_{t-1}(i)e_{k-1}(\boldsymbol{w}_{t-1,-i})}{e_k(\boldsymbol{w}_{t-1})}$. Hence, the marginal inclusion probabilities can be expressed in terms of symmetric polynomials that can be efficiently computed in $O(N\log^2(k))$ time via Fast Fourier Transform methods (see, *e.g.*, Shpilka and Wigderson (2001)). Further speedup is possible by exploiting the fact that the weight of only one of the components change at a round. This faster iterative method is derived in Section 11.2 of the supplementary material.

3. Sampling the predicted set: Upon computing the marginal inclusion probabilities, we use Madow's systematic sampling scheme outlined in Algorithm 2 to sample a k-set. The overall prediction policy is summarized in Algorithm 3.

Algorithm 3 k-sets via SAGE with $\pi_{\text{base}} =$ Hedge

Input: $\boldsymbol{w} \leftarrow \mathbf{1}$, learning rate $\eta > 0$.

1: **for** every time t **do**
2: $\quad \boldsymbol{w}_i \leftarrow \boldsymbol{w}_i \exp(\eta \mathbb{1}(y_{t-1} = i)), \forall i \in [N]$.
3: $\quad p(i) \leftarrow \frac{w(i)e_{k-1}(\boldsymbol{w}_{-i})}{e_k(\boldsymbol{w})}, \forall i \in [N]$,
4: $\quad$ Sample a k-set with the marginal inclusion probabilities $\boldsymbol{p}$ using Algorithm 2.
5: **end for**

4.1.1 Regret Bounds

Recall that, in expectation, the performance of Algorithm 3 and the base policy Hedge are identical. It is well-known that by adaptively tuning the learning rate η, the Hedge policy with n experts admits the following data-dependent small-loss regret bound (Koolen et al., 2010; Erven et al., 2011)

$$\text{Regret}_T \le \sqrt{2l_T^* \ln n} + \ln n, \quad (13)$$

where l_T^* denotes the cumulative loss incurred by the best fixed expert in hindsight for the given loss matrix. In the case of the k-sets problem, the total number of experts is given by $n = \binom{N}{k} \le (\frac{Ne}{k})^k$. Hence, the SAGE prediction framework with Hedge as the base policy yields the following adaptive regret bound:

$$\text{Regret}_T(\boldsymbol{y}) \le \sqrt{2kl_T^*(\boldsymbol{y})\ln(Ne/k)} + k\ln(Ne/k), \quad (14)$$

where $l_T^*(\boldsymbol{y})$ is the number of mistakes incurred by the best fixed k-set in hindsight for the sequence $\boldsymbol{y}$. Since $l_T^*(\boldsymbol{y}) \le T$, the regret upper bound (14) is sublinear in the horizon-length. However, the bound could be much smaller if the offline oracle incurs a small number of mistakes for a particular sequence.

Discussion: Algorithm 3 offers a new projection and decomposition-free approach to break the existing $O(N^2)$ complexity barrier for the k-sets problem (Herbster and Warmuth, 2001). The work by Uchiya et al. (2010) studies a bandit version of the k-sets problem and proposes **Exp3.M** policy, which incurs $O(\sqrt{kNT\log N/k})$ regret. However, this bound cannot be compared with our (smaller) regret bound, applicable in the full-information setting. Furthermore, they use *dependent rounding* method, which is more complex than Madow's sampling that we use here.

4.2 k-sets with FTRL

It is also possible to design efficient online policies for the k-sets problem with a base policy other than Hedge. In Section 11.3 of the supplementary, we show how the standard *Follow-the-Regularlized-Leader* (FTRL) framework can be augmented with the systematic sampling schemes to design an efficient online prediction policy for a generalized version of the

k-experts problem with the sum-reward function. A drawback of the FTRL approach is that, unlike Hedge, this policy does not admit an adaptive regret bound. Due to space constraints, we defer the detailed discussion to Section 11.3 of the supplementary material.

5 k-experts WITH Pairwise-rewards

In this section, we design an online prediction policy for a special case of the k-experts problem with the pairwise-reward function and binary rewards (see Table 1)[1]. Recall that, in the k-sets problem, the adversary chooses a single item at each round (so that only one component of the reward vector $\boldsymbol{r}_t$ is one and the rest are zero). On the contrary, in this problem, the adversary secretly selects a *pair* of items at each round (so that exactly two components of the reward vector $\boldsymbol{r}_t$ are one and the rest are zero). If *both* the items chosen by the adversary are included in the predicted k-set, the learner receives a unit reward; else, it receives zero rewards for that round. The following hardness result is immediate.

Proposition 1 *The offline version of the k-experts problem with pairwise-rewards is **NP-Hard**.*

Proof: The proof follows from a simple reduction of the **NP-Hard** Densest k-subgraph problem (Sotirov, 2020) to the offline optimization problem. Consider an arbitrary graph $\mathcal{G}$ on N vertices and T edges denoted by $e_1, e_2, \ldots, e_T$. Construct an instance of the k-experts problem with pairwise-rewards such that, at round t, the adversary chooses the pair of items corresponding to the vertices of the edge $e_t, 1 \leq t \leq T$. Then the problem of finding a subgraph of k vertices such that the number of edges in the induced subgraph is maximum (*i.e.,* the Densest k-subgraph of $\mathcal{G}$) reduces to the offline problem of selecting the most rewarding k items to maximize the cumulative reward in the k-experts problem with pairwise-rewards. □

In principle, we can use the SAGE framework to obtain the optimal pairwise inclusion probabilities and then sample k items accordingly. However, there are two main difficulties with this approach - (1) unlike Eqn. (2), there is no known succinct characterization of the feasible set of pairwise inclusion probability vector when k items are chosen from N items without replacement, and (2) given a feasible pairwise inclusion probability vector, it is not known how to efficiently sample k items accordingly. The above roadblocks are not surprising given the hardness of the offline problem. This prompts us to propose the following approximate policy described in Algorithm 4.

[1]The general case with arbitrary rewards can be handled using a similar FTRL approach as in Section 4.2.

Algorithm 4 Algorithm for pairwise-rewards

1: Treat each pair of items as a single *super-item*.
2: Use SAGE to sample k distinct super-items from $\binom{N}{2}$ super-items per round.

Since any particular item may be a part of $k-1$ super-items, it is possible that the set of sampled super-items in Algorithm 4 includes an item multiple times. However, it is easy to see that the number of items contained in the union of any k super-items is bounded between $\sqrt{2k}$ and $2k$. Hence, replacing N with $\binom{N}{2}$ (the number of super-items) in Eqn. (14) yields the following performance guarantee for Algorithm 4:

Offline oracle reward with at most $\sqrt{2k}$ items - the reward accrued by Algorithm 4 with at most $2k$ items is upper bounded by:

$$2\sqrt{kl_T^* \ln(N^2 e/2k)} + 2k \ln(N^2 e/2k),$$

where l_T^* is the loss incurred by the optimal offline oracle using $2k$ items. Algorithm 4 is an instance of *improper learning* algorithm where the online policy competes with a weaker oracle.

6 LEARNING POLICIES FOR MONOTONE REWARDS

In this section, we use the SAGE framework to design an efficient online policy to learn any smooth monotone reward function. Recall that a set function $f : 2^{[N]} \to \mathbb{R}$ is *monotone* if $f(S_1) \geq f(S_2), \forall S_2 \subseteq S_1 \subseteq [N]$. A set function f is *modular* if for any subset $S \subseteq [N]$, we have: $f(S) = \sum_{i \in S} f(\{i\})$. Our starting point is the following fundamental result, which approximates *any* set function by modular functions.

Theorem 3 (Iyer and Bilmes (2012)) *For a given set X and any set function $f : 2^X \to \mathbb{R}$ and any set $Y \subseteq X$, there are two modular functions $m_u : 2^X \to \mathbb{R}$ and $m_l : 2^X \to \mathbb{R}$ such that $m_l \leq f \leq m_u$ and $m_l(Y) = f(Y) = m_u(Y)$. Furthermore, the functions m_l and m_u can be expressed explicitly in terms of the function f.*

See Appendix 11.4 for the expressions of approximating modular functions and other computational details. We assume that the reward function f_t, chosen by the adversary at any round $t \in [T]$, is monotone with $f_t(\emptyset) = 0, \forall t \in [T]$. We also assume that the reward functions are "smooth", i.e., there exists a finite constant G such that $\forall S \subseteq [N], x \in [N]$, we have:

$$|f_t(S) - f_t(S \setminus \{x\})| \leq G, \ \forall t \geq 1. \tag{15}$$

In the k-experts setting, the online prediction policy can select only a subset of k experts at each round.

We consider an improper learning setup where our objective is to design a prediction policy that attains at least a k/N fraction of the total cumulative rewards obtained by taking *all* N experts at each round up to an $O(\sqrt{T})$ term. Note that the comparator in this section is different from that of the standard regret metric (1), where the reward accrued by the online policy is compared against the optimal k-set in hindsight.

Using Theorem 3, we can construct a modular set function m_l^t corresponding to the function f_t such that:

$$f_t \geq m_l^t, \qquad \text{and} \qquad f_t([N]) = m_l^t([N]). \tag{16}$$

Consider a `sum-reward` variant of the k-`sets` problem, where the reward $g_t(i)$ for the i^{th} expert at round t is set to be equal to $m_l^t(\{i\}), i \in [N]$. We now use a prediction policy that minimizes the static regret (1) with respect to the linearized reward vectors $\{\boldsymbol{g}_t\}_{t\geq 1}$:

$$\mathcal{R}_T = \max_{p^* \in \Delta_N^k} \sum_{t\leq T} \langle g_t, p^* \rangle - \sum_{t \leq T} \langle g_t, p_t \rangle. \tag{17}$$

From Eqn. (43) of the Supplementary, it follows that the `FTRL` (η) policy with entropic regularizer guarantees the following regret bound for the `sum-reward` problem:

$$\mathcal{R}_T \leq \frac{k \ln(N/k)}{\eta} + 2\eta \sum_{t\leq T} \left\| \boldsymbol{g}_t^2 \right\|_{k,\infty},$$

where $\left\| \boldsymbol{x}^2 \right\|_{k,\infty}$ denotes the sum of the k largest components of the vector $(x_1^2, x_2^2, \ldots, x_N^2)$. Using the smoothness assumption (15), we show in Appendix 11.4 that $\left\| \boldsymbol{g}_t^2 \right\|_{k,\infty} \leq B^2$, where $B = O(GN^{3/2}\sqrt{k})$ for arbitrary reward functions. We also show that the bound can be improved to $B = O(G\sqrt{k})$ for submodular functions. Hence, with the optimal tuning of the learning rate η, the `FTRL` policy achieves the following regret bound:

$$\mathcal{R}_T \leq 2B\sqrt{2kT\ln(N/k)}. \tag{18}$$

Now observe that:

$$\begin{aligned} \mathbb{E}[f_t(S_t)] &= \sum_{S_t} p_t(S_t) f_t(S_t) \geq \sum_{S_t} p_t(S_t) m_l^t(S_t) \\ &= \sum_{i=1}^{N} p_t(i) g_t(i) = \langle g_t, p_t \rangle. \end{aligned} \tag{19}$$

Furthermore, we also have:

$$\sum_{t\leq T} f_t([N]) \overset{(a)}{=} \sum_{t\leq T} \sum_{i=1}^{N} g_t(i) \leq \frac{N}{k} \max_{p^* \in \Delta_N^k} \sum_{t\leq T} \langle g_t, p^* \rangle, \tag{20}$$

where we have used Eqn. (16) in Eqn. (a). Substituting the bounds from Eqn. (19) and (20) into the regret bound (17) yields the following performance guarantee:

$$\frac{k}{N} \sum_{t\leq T} f_t([N]) - \sum_{t\leq T} \mathbb{E}[f_t(S_t)] \leq 2B\sqrt{2kT\ln(N/k)}.$$

Hence, for arbitrary monotone reward functions, the prediction policy asymptotically achieves a k/N fraction of the maximum possible cumulative reward.

7 LOWER BOUNDS

In this section, we lower bound the achievable regret for different variants of the k-`experts` problem. To begin with, consider the setting where the adversary chooses binary rewards with exactly one non-zero reward per round. In this setting, Bhattacharjee et al. (2020) established the following regret lower bound for the `Sum-reward` variant of the k-`experts` problem:

Theorem 4 (Regret Lower bound for `Sum-reward`) *For any online policy with $\frac{N}{k} \geq 2$ and $T \geq 1$, we have*

$$\mathcal{R}_T^{\textit{Sum-reward}} \geq \sqrt{\frac{kT}{2\pi}} - \Theta(\frac{1}{\sqrt{T}}).$$

Note that with the above rewards structure, the `Sum-reward`, the `Max-reward`, and the `Pairwise-reward` variants of the k-`experts` problem become identical. Hence, Theorem 4 also yields a lower bound to all of the above variants of the k-`experts` problem. However, from the standard `Hedge` achievability bound applied to the meta-experts (Eqn. (14)), it can be readily observed that the upper and lower regret bounds differ by a logarithmic factor. Our main result in this section is the following tight regret lower bound for the `Max-Reward` variant of the k-`experts` problem, that removes the above logarithmic gap.

Theorem 5 (Regret Lower Bound for `Max-reward`) *For any online policy with $T \geq 16k\ln(\frac{N}{k})$ and $\frac{N}{k} \geq 7$, we have*

$$\mathcal{R}_T^{\textit{Max-reward}} \geq 0.02\sqrt{kT\ln\frac{N}{k}}.$$

Compared to the standard lower bounds (Cesa-Bianchi and Lugosi, 2006), a distinguishing feature of the above regret lower bound is its non-asymptotic nature.

Proof outline: The proof utilizes the standard probabilistic technique where the worst-case regret is lower bounded by the average regret over an ensemble of k-`experts` problems. However, the analysis becomes complex as the reward accrued at each round t is a non-linear function of the reward vector. To alleviate this difficulty, we first partition the pool of N experts into k disjoint subsets. Then we select the cumulative best expert in hindsight from each subset in order to lower bound the optimal offline reward. Please refer to Section 11.5 in the supplementary material for detailed proof.

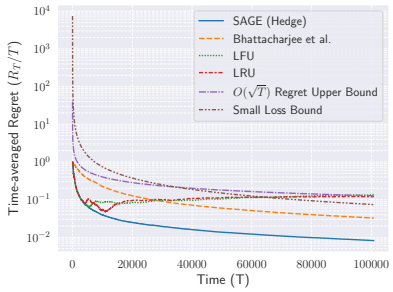

Figure 1: Comparison among different k-`set` policies with $k/N = 0.1, N \sim 2400$ for the MovieLens Dataset.

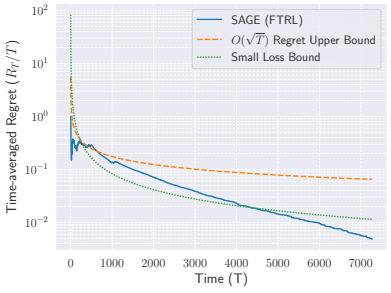

Figure 2: Performance of the `SAGE` policy for pairwise predictions with $k/N = 0.02$ for the Reality Mining Dataset.

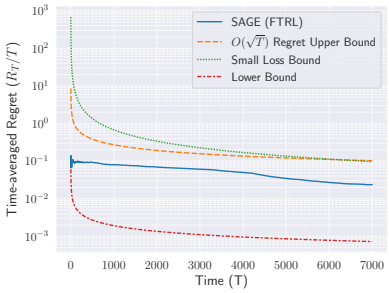

Figure 3: Performance of `SAGE` for the `Max-Reward` function, with $k/N = 0.01$ for the MovieLens Dataset.

8 NUMERICAL EXPERIMENTS

k-`sets`: [2] Assume that there is a collection of N movies. The user may request any of the N movies at each round. The learner sequentially predicts (possibly randomly) a set of k movies that the user is likely to watch at a given round. At each round, the learner receives a unit reward if the movie chosen by the user is in the predicted set; else, it receives zero rewards for that round. The learner's goal is to maximize the total number of correct predictions over a given time interval. In our experiments, we use the MovieLens 1M dataset (Harper and Konstan, 2015) for generating the sequence of movies chosen by the user. The dataset contains $T \sim 10^5$ ratings for $N \sim 2400$ movies along with the timestamps. We assume that a user rates a movie immediately after watching it. The plot in Figure 1 compares the normalized regrets of the proposed `SAGE` policy (with $\pi_{\text{base}} =$ `Hedge`), the `FTPL` policy proposed by Bhattacharjee et al. (2020), and two other baseline prediction policies - LFU and LRU, which treat the prediction problem as a paging problem (Geulen et al., 2010). From the plot, it is clear that the `SAGE` policy decisively outperforms all other policies.

k-`experts` with `Pairwise-reward`: In our next experiment, we use the MIT Reality Mining dataset (Eagle and Pentland, 2006) to understand the efficacy of the prediction policy for pairwise rewards proposed in Section 5. The dataset contains timestamped human contact data among 100 MIT students collected using standard Bluetooth-enabled mobile phones over 9 months. In our experiments, we consider a subset of $N = 20$ students with $\binom{20}{2} = 190$ potential contact pairs. The learner's task is to predict a sequence of k-sets that include both the students involved in the contact for each timestamp. As described in Section 5, we design an approximate prediction policy by considering each pair of students as a *super-item* and use the `SAGE` framework with $\pi_{\text{base}} =$ `FTRL`. The normalized regret achieved by this policy is shown in Figure 2. To compute the optimal static offline reward, we used a brute-force search. From the plots, we see that the normalized regret of this policy approaches zero for long-enough time-horizon.

k-`experts` with `Max-reward`: In our final experiment, we use a subset of the MovieLens dataset with $T \sim 7000$ ratings for $N = 200$ movies. We assume that the movies are sorted according to genres so that if the movie i is chosen by the user at each round, the learner receives a reward of $\max_{j \in S} \left(1 - \frac{1}{N}|j - i|\right)$ for predicting the set S. This reward function roughly emulates the practical requirement that if the requested movie is not in the predicted set, then it is preferable to recommend a similar movie than a completely different one. In Figure 3, we plot the normalized regret of the `SAGE` policy with $\pi_{\text{base}} =$ `FTRL`, along with the lower bound given in Theorem 5. From the plot, we can see that the normalized regret shows a downward trend with T even with the `FTRL` policy, albeit there is a non-trivial gap with the lower bound. This gap is expected as the `FTRL` policy is optimal for the `Sum-reward` function, but not necessarily so for the `Max-reward` function. Please see Section 12 of the supplement for additional results.

9 CONCLUSION

In this paper, we formulated the k-`experts` problem and designed efficient learning policies for some of its variants using the `SAGE` framework. We also derived a tight regret lower bound for the `Max-reward` variant and characterized the set of all mistake bounds for the k-`sets` problem achievable by online policies. In the future, it would be interesting to benchmark the performance of the algorithms on larger datasets.

[2] All codes used in the experiments are available at: https://github.com/sourav22899/k-sets-problem.

10 ACKNOWLEDGMENTS

This work is partially supported by the grant IND-417880 from Qualcomm (USA) and a research grant from the Govt. of India under the Institutes of Eminence (IoE) initiative. The computational results reported in this work were performed on the AQUA Cluster at the High Performance Computing Environment of IIT Madras.

References

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Vladimir Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56(2):153–173, 1998.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. Technical report, Carnegie-Mellon Univ. Pittsburgh PA School of Computer Science, 2007.

Wouter M Koolen, Manfred K Warmuth, and Jyrki Kivinen. Hedging structured concepts. In *COLT*, pages 93–105. Citeseer, 2010.

Thomas M. Cover. Behavior of sequential predictors of binary sequences. *Transactions of the Fourth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes, Prague*, pages 263–272, 1966. URL https://isl.stanford.edu/people/cover/papers/paper3.pdf.

Alexander Rakhlin and Karthik Sridharan. A tutorial on online supervised learning with applications to node classification in social networks. *arXiv preprint arXiv:1608.09014*, 2016.

Mark Herbster and Manfred K Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1(281-309):10–1162, 2001.

Manfred K Warmuth and Dima Kuzmin. Randomized online pca algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9(Oct):2287–2320, 2008.

Daiki Suehiro, Kohei Hatano, Shuji Kijima, Eiji Takimoto, and Kiyohito Nagano. Online prediction under submodular constraints. In *International Conference on Algorithmic Learning Theory*, pages 260–274. Springer, 2012.

Eiji Takimoto and Kohei Hatano. Efficient algorithms for combinatorial online prediction. In *International Conference on Algorithmic Learning Theory*, pages 22–32. Springer, 2013.

Amit Daniely and Yishay Mansour. Competitive ratio vs regret minimization: achieving the best of both worlds. In *Algorithmic Learning Theory*, pages 333–368. PMLR, 2019.

Rajarshi Bhattacharjee, Subhankar Banerjee, and Abhishek Sinha. Fundamental limits on the regret of online network-caching. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4 (2):1–31, 2020.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3:71–104, 2014.

Nicholas Harvey, Christopher Liaw, and Tasuku Soma. Improved algorithms for online submodular maximization via first-order regret bounds. *Advances in Neural Information Processing Systems*, 33, 2020.

Alon Cohen and Tamir Hazan. Following the perturbed leader for online structured learning. In *International Conference on Machine Learning*, pages 1034–1042. PMLR, 2015.

Yves Tillé. Some remarks on unequal probability sampling designs without replacement. *Annales d'Economie et de Statistique*, pages 177–189, 1996.

Herman Otto Hartley. Systematic sampling with unequal probability and without replacement. *Journal of the American Statistical Association*, 61(315):739–748, 1966.

Muhammad Hanif and KRW Brewer. Sampling with unequal probabilities without replacement: a review. *International Statistical Review/Revue Internationale de Statistique*, pages 317–335, 1980.

William G Madow et al. On the theory of systematic sampling, ii. *The Annals of Mathematical Statistics*, 20(3):333–354, 1949.

Wouter M Koolen and Tim Van Erven. Second-order quantile methods for experts and combinatorial games. In *Conference on Learning Theory*, pages 1155–1175. PMLR, 2015.

Tim Erven, Wouter M Koolen, Steven Rooij, and Peter Grünwald. Adaptive hedge. *Advances in Neural Information Processing Systems*, 24:1656–1664, 2011.

Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001.

Taishi Uchiya, Atsuyoshi Nakamura, and Mineichi Kudo. Algorithms for adversarial bandit problems with multiple plays. In *International Conference on Algorithmic Learning Theory*, pages 375–389. Springer, 2010.

Renata Sotirov. On solving the densest k-subgraph problem on large graphs. *Optimization Methods and Software*, 35(6):1160–1178, 2020.

Rishabh Iyer and Jeff Bilmes. Algorithms for approximate minimization of the difference between submodular functions, with applications. *arXiv preprint arXiv:1207.0560*, 2012.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Sascha Geulen, Berthold Vöcking, and Melanie Winkler. Regret minimization for online buffering problems using the weighted majority algorithm. In *COLT*, pages 132–143. Citeseer, 2010.

Nathan Eagle and Alex Pentland. Reality Mining: Sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, 2006.

Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *NIPS*, volume 23, pages 586–594, 2010.

Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*, pages 2071–2080. PMLR, 2017.

Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

José M Amigó, Sámuel G Balogh, and Sergio Hernández. A brief review of generalized entropies. *Entropy*, 20(11):813, 2018.

A.A. Fedotov, P. Harremoes, and F. Topsoe. Refinements of pinsker's inequality. *IEEE Transactions on Information Theory*, 49(6):1491–1498, 2003. doi: 10.1109/TIT.2003.811927.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Wei-Li Wu, Zhao Zhang, and Ding-Zhu Du. Set function optimization. *Journal of the Operations Research Society of China*, 7(2):183–193, 2019.

Mukund Narasimhan and Jeff A Bilmes. A submodular-supermodular procedure with applications to discriminative structure learning. *arXiv preprint arXiv:1207.1404*, 2012.

Pascal Massart. Concentration inequalities and model selection. 2007.

Spencer Greenberg and Mehryar Mohri. Tight lower bound on the probability of a binomial exceeding its expectation. *Statistics & Probability Letters*, 86: 91–98, 2014.

Daniel S Berger, Nathan Beckmann, and Mor Harchol-Balter. Practical bounds on optimal caching with variable object sizes. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2 (2):1–38, 2018.

Supplementary Material: k-experts - Online Policies and Fundamental Limits

11 Proofs and Derivations

11.1 Proof of Theorem 2

Phase-I: Computation of the Marginal Inclusion Probabilities $\boldsymbol{p}_t$: Similar to the treatment in Rakhlin and Sridharan (2016), we use a potential function-based argument to derive a set of marginal inclusion probabilities at each time t that leads to the loss function $\phi(\cdot)$. Let $\{\phi_t : [N]^t \to [0,1]\}_{t=0}^T$ be a sequence of potential functions satisfying the boundary condition

$$\phi_T(\boldsymbol{y}) = \phi(\boldsymbol{y}). \tag{21}$$

We define ϕ_0 to be a suitable constant. In order to achieve the loss function $\boldsymbol{\phi}(\cdot)$, we require the following equality to be valid for all sequences $\boldsymbol{y} \in [N]^T$:

$$\mathbb{E}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}(y_t \notin S_t)\right) = \sum_{t=1}^{T}\left(\phi_t(\boldsymbol{y}^t) - \phi_{t-1}(\boldsymbol{y}^{t-1})\right) + \phi_0, \tag{22}$$

where the above equation follows from telescoping the summation and using the boundary condition (21). For a given initial segment of the sequence $\boldsymbol{y}^{t-1}$, consider an online policy that includes the i^{th} element in the predicted set S_t with the conditional probability $p_{ti}(\boldsymbol{y}^{t-1})$. Clearly

$$\mathbb{P}(y_t \notin S_t | \boldsymbol{y}^{t-1}) = 1 - \sum_{i=1}^{N} p_{ti}(\boldsymbol{y}^{t-1})\mathbb{1}(y_t = i). \tag{23}$$

Hence, combining equations (22) and (23), the achievability is ensured if we can exhibit a sequence of potential functions $\{\phi_t(\cdot)\}$ and a randomized online strategy for selecting the sets S_t, such that the following equality holds for every sequence $\boldsymbol{y} \in [N]^T$:

$$\sum_{t=1}^{T}\left(-\sum_{i=1}^{N}\frac{p_{ti}(\boldsymbol{y}^{t-1})\mathbb{1}(y_t = i)}{T} + \phi_{t-1}(\boldsymbol{y}^{t-1}) - \phi_t(\boldsymbol{y}^t) + \frac{1}{T}(1-\phi_0)\right) = 0. \tag{24}$$

We now consider the following candidate sequence of potential functions:

$$\phi_t(\boldsymbol{y}_t) \equiv \mathbb{E}\phi(\boldsymbol{y}_t, \epsilon_{t+1}^T), \quad \forall t, \tag{25}$$

where the expectation is taken over a random sequence $\boldsymbol{\epsilon}_{t+1}^T$ such that each component $\epsilon_j, t+1 \le j \le N$ is distributed i.i.d. uniformly over the set $[N]$. It is easy to see that, the boundary condition (21) is satisfied. Furthermore, from the condition given in the statement of the theorem, we have $\phi_0 = \mathbb{E}\phi(\boldsymbol{\epsilon}_1^T) = 1 - k/N$. Next, we exhibit a prediction strategy with inclusion probabilities $\{p_{ti}(\boldsymbol{y}^{t-1})\}$ such that the equation (24) is satisfied. For, this, we set each of the terms of the equation (24) identically to zero for any sequence $\boldsymbol{y} \in [N]^T$. This yields the following conditional inclusion probability of the i^{th} element for any initial segment of the request sequence $\boldsymbol{y}^{t-1} \in [N]^{t-1}$:

$$p_{ti}(\boldsymbol{y}^{t-1}) = T\left(\phi_{t-1}(\boldsymbol{y}^{t-1}) - \phi_t(\boldsymbol{y}^{t-1}i)\right) + \frac{k}{N}, \quad \forall i \in [N]. \tag{26}$$

From the definition (25), we have that $\frac{1}{N}\sum_{i=1}^{N}\phi_t(\boldsymbol{y}^{t-1}i) = \phi_{t-1}(\boldsymbol{y}^{t-1})$. Hence, summing equation (26) over all $i \in [N]$, we have

$$\sum_{i=1}^{N} p_{ti}(\boldsymbol{y}^{t-1}) = k.$$

Thus, the scalars $\{\boldsymbol{p}_{ti}\}_{i=1}^N$ satisfy the requirement in equation (2). Hence, to guarantee that Eqn. (26) yields a valid prediction strategy, we only need to ensure that $0 \leq p_{ti} \leq 1, \forall i \in [N]$. In the following, we show that this requirement is also satisfied, thanks to the stability property of the loss function $\phi(\cdot)$. For this, we are required to ensure the following bound for all $\boldsymbol{y}^{t-1}$:

$$-\frac{k}{N} \leq T\left(\frac{1}{N}\sum_{i=1}^{N}\phi_t(\boldsymbol{y}^{t-1}i) - \phi_t(\boldsymbol{y}^{t-1}i)\right) \leq 1 - \frac{k}{N}. \tag{27}$$

It immediately follows that the stability conditions, given by equations (6) and (7), are sufficient to ensure the bound in Eqn. (27).

Phase-II: Sampling the Predicted set We use the conditional marginal inclusion probabilities $\boldsymbol{p}_t$, derived in Eqn. (26), to construct a consistent randomized output set S_t with $|S_t| = k$. Since the inclusion probabilities satisfy the feasibility constraints, we can use the Algorithm 2 to construct the predicted set. Phase-I and Phase-II, taken together, complete the proof of the theorem.

11.2 Iterative evaluation of the marginal inclusion probabilities

At any time t, consider the formal power series $g_t(X)$ defined as

$$g_t(X) = \prod_{i\in[N]}(X - w_t(i)) = \sum_{j=0}^{N} a_{tj}X^j, \tag{28}$$

i.e., $\forall j = 0, \cdots, N,\ a_{tj}$ is the coefficient of X^j in the expansion of $g_t(X)$. Then, by Vieta's formulae, we obtain,

$$\begin{aligned}\sum_{1\leq i_1<i_2<\cdots<i_k\leq N}\prod_{j=1}^{k} w_t(i_j) &= (-1)^k a_{t,N-k}\\ \iff \sum_{S'\subset[N]:|S'|=k} w_t(S') &= (-1)^k a_{t,N-k}.\end{aligned} \tag{29}$$

Now define

$$g_{ti}(X) = \frac{g_t(X)}{X - w_t(i)} = \sum_{j=0}^{N-1} b_{tj}^{(i)}X^j, \tag{30}$$

where $b_{tj}^{(i)}$ is the coefficient of X^j in the expansion of $g_{ti}(X)$. Again using Vieta's formula, it follows that,

$$\sum_{S\subset[N]\setminus\{i\}:|S|=k-1} w_t(S) = (-1)^{k-1} b_{t,N-k}^{(i)}. \tag{31}$$

Therefore, it follows that the probability selection rule (10) can be expressed as below:

$$p_t(i) = -\frac{w_{t-1}(i)b_{t-1,N-k}^{(i)}}{a_{t-1,N-k}},\ \forall i \in [N]. \tag{32}$$

It now remains to find a computationally efficient way of updating the coefficients $a_{tj}, b_{tj}^{(i)}$. To this direction, given the coefficients $\{a_{tj}\}_{j=0}^N$, we compute the coefficients $\{b_{tj}\}_{j=0}^{N-1}$ in the following way. Using the formal power series expansion $(1-X)^{-1} = \sum_{l\geq 0} X^l$, one can write,

$$\begin{aligned} g_{ti}(X) &= -w_t^{-1}(i)g_t(X)\sum_{l\geq 0} X^l w_t^{-l}(i)\\ &= -w_t^{-1}(i)\sum_{j=0}^{N}\sum_{l=0}^{\infty} a_{tj}w_t^{-l}(i)X^{j+l}. \end{aligned} \tag{33}$$

Therefore, $\forall 0 \leq j \leq N-1$,

$$b_{tj}^{(i)} = -\sum_{l=0}^{j} a_{tl} w_t^{-(j-l+1)}(i) \tag{34}$$

Consequently, we can further express the probability selection rule from Eq. (32) as

$$p_t(i) = \frac{\sum_{j=0}^{N-k} a_{t-1,j} w_{t-1}^{-(N-k-j)}(i)}{a_{t-1,N-k}}. \tag{35}$$

We now proceed to find update rule for the coefficients a_{tj}. Let f_t be the file requested at time t. Then, $R_t(f_t) = R_{t-1}(f_t) + \rho_t$, where $\rho_t = \mathbb{1}(f_t \in S_t)$, whereas, $R_t(i) = R_{t-1}(i)$ if $i \neq f_t$. Therefore,

$$\begin{aligned} g_t(X) &= \prod_{i=1}^{N}(X - w_t(i)) = g_{t-1}(X) \cdot \frac{X - w_t(f_t)}{X - w_{t-1}(f_t)} \\ &= g_{t-1,f_t}(X)(X - w_t(f_t)) = \sum_{j=0}^{N-1} b_{t-1,j}^{(f_t)} X^j (X - w_t(f_t)). \end{aligned} \tag{36}$$

Therefore, using the above and the update rule of $b_{tj}^{(i)}$ from Eq. (34), we obtain,

$$\begin{aligned} a_{tj} &= b_{t-1,j-1}^{(f_t)} - w_t(f_t) b_{t-1,j}^{(f_t)} \\ &= w_t(f_t) \sum_{k=0}^{j} a_{t-1,k} w_{t-1}^{-(j-k+1)}(f_t) - \sum_{k=0}^{j-1} a_{t-1,k} w_{t-1}^{-(j-k)}(f_t) \\ &= (e^{\eta} - 1) \sum_{k=0}^{j} a_{t-1,k} w_{t-1}^{-(j-k)}(f_t) + a_{t-1,j}, \end{aligned} \tag{37}$$

where in the last step we have used the fact that $w_t(f_t) w_{t-1}^{-1}(f_t) = e^{\eta}$, since f_t is the requested file at time t and hence $R_t(f_t) = R_{t-1}(f_t) + 1$. The update Eq. (37) can be used to obtain a further simplified recurrence to the update of the coefficients $a_{t,i}$ as below:

$$\begin{aligned} a_{t,j} &= (e^{\eta} - 1) w_{t-1}^{-1}(f_t) \sum_{k=0}^{j-1} a_{t-1,k} w_{t-1}^{-(j-1-k)}(f_t) + e^{\eta} a_{t-1,j}, \\ &= w_{t-1}^{-1}(f_t)\big(a_{t,j-1} - a_{t-1,j-1}\big) + e^{\eta} a_{t-1,j}, \ \forall 1 \leq j \leq N, \end{aligned} \tag{38}$$

$$a_{t,0} = e^{\eta} a_{t-1,0}. \tag{39}$$

Using the update equations of $\{a_{tj}\}_{j=1}^{N}$ and $\{p_t(j)\}$ from Eqs. (38), (39) and (35) respectively, we have the following iterative numerical procedure for computing the marginal inclusion probabilities:

11.3 Generalized k-sets with FTRL

In this section, we design an efficient online policy for a generalized version of the k-sets problem where the reward per round is modulated using a non-decreasing concave function $\psi : \mathbb{R}_{\geq \mathbf{0}} \to \mathbb{R}$, called the *link function*. In particular, the reward of the learner at round t is defined to be $\psi(\boldsymbol{r}_t \cdot \boldsymbol{p}_t)$. In the special case when $\psi(\cdot)$ is the identity function, we recover the standard k-sets problem. The notion of link functions is common in the literature on Generalized Linear Models (Filippi et al., 2010; Li et al., 2017). Note that, although the reward function could be non-linear, it still depends only on the marginal inclusion probabilities of the elements, and hence the SAGE framework applies. Formally, the objective of the learner is to design an efficient online learning policy to minimize the *static regret* with respect to an offline oracle (the best fixed k-set in the hindsight), *i.e.*,

$$\mathcal{R}_T := \max_{\boldsymbol{p}^* \in \Delta(\mathcal{C}_k^N)} \sum_{t=1}^{T} \psi(\boldsymbol{r}_t \cdot \boldsymbol{p}^*) - \sum_{t=1}^{T} \psi(\boldsymbol{r}_t \cdot \boldsymbol{p}_t), \tag{40}$$

Algorithm 5 Iterative Computation of the Marginal Inclusion Probabilities

Input: Learning rate $\eta > 0$,
Initialize: $\boldsymbol{R}_0 = \mathbf{0}, a_{0,j} = (-1)^{N-j}\binom{N}{j},\ \forall 0 \le j \le N.$

1: **for** $t = 1, \cdots, T$ **do**

2: Compute $w_{t-1}(i) = \exp(\eta R_{t-1}(i))\ \forall i \in [N]$, and set $p_t(i) = \frac{\sum_{j=0}^{N-k} a_{t-1,N-k-j} w_{t-1}^{-j}(i)}{a_{t-1,N-k}},\ \forall i \in [N]$.

3: Sample a set $S_t \subset [N]$ with $|S_t| = k$ according to Madow's systematic sampling using the probabilities $\{p_t(i)\}_{i\in[N]}$ and construct the vector $\boldsymbol{y}_t \in \{0,1\}^N$, such that $y_{t,i} = 1(i \in S_t)$.

4: Observe the requested file index f_t and update

$$R_t(i) \leftarrow R_{t-1}(i) + \mathbb{1}(f_t = i).$$

5: Update

$$a_{t,0} \leftarrow e^{\eta} a_{t-1,0}$$
$$a_{t,j} \leftarrow w_{t-1}^{-1}(f_t)\big(a_{t,j-1} - a_{t-1,j-1}\big) + e^{\eta} a_{t-1,j},\ 1 \le j \le N.$$

6: **end for**

We augment the well-known *Follow-the-Regularized-Leader* (**FTRL**) framework with the Systematic Sampling scheme in Algorithm 2 to design an efficient online policy for the generalized ***k*-sets** problem with a sublinear regret. Interestingly, we will see that, when specialized to the ***k*-sets** problem, the **FTRL**-based approach yields a different policy from the `Hedge`-based Algorithm 3. The problem of finding the optimal marginal inclusion probabilities to minimize the regret in Eqn. (40) is an instance of the Online Convex Optimization (OCO) problem (Hazan, 2019). We use the standard *Follow-the-Regularized-Leader* (**FTRL**) paradigm to design an online prediction policy with sublinear regret. We refer the reader to Hazan (2019) for an excellent introduction to the **OCO** framework in general, and the **FTRL** policy in particular.

Recall that, in the general **FTRL** paradigm, the learner's action at time t is obtained by maximizing the sum of the cumulative rewards (or a linear lower bound to it) upto time $t-1$ and a strongly concave regularizer $g : \Omega \to \mathbb{R}$, where Ω is the set of all feasible actions of the learner. For the **Generalized *k*-sets** problem, the vector of marginal inclusion probabilities is constrained to be in the set $\Omega = \Delta_k^N$, where $\Delta_k^N = \{\boldsymbol{p} \in [0,1]^N : \sum_{i=1}^N p_i = k.\}$ In the following, we choose the usual (Shannon) entropic regularizer as our regularization function, *i.e.*, we take $g(\boldsymbol{p}) = -\sum_{i=1}^N p_i \ln p_i$. This choice is motivated by the well-known fact that the entropic regularization yields the `Hedge` policy for the `Experts` problem (where $k = 1$) (Hazan, 2019). In our numerical experiments, we also investigate the performance of the Rényi and Tsallis entropic regularizers of various orders (Amigó et al., 2018). Choosing the entropic regularizer leads to the following convex program for determining the marginal inclusion probabilities $\boldsymbol{p}_t$ at the t^{th} round:

$$\boldsymbol{p}_t = \arg\max_{\boldsymbol{p}\in\Delta_k^N} \left[\Big(\sum_{s=1}^{t-1} \nabla_s\Big)^T \boldsymbol{p} - \frac{1}{\eta}\sum_{i=1}^N p_i \ln p_i, \right] \tag{41}$$

where $\nabla_{s,i} \equiv r_{s,i}\psi'(\boldsymbol{r}_s^T \boldsymbol{p}_s)$ denotes the i^{th} component of the gradient vector. Using convex duality, the optimal solution to (41) may be quickly determined in $\tilde{O}(N)$ time as shown in Algorithm 6 below.

See Section 11.3.1 below for the derivation of the Algorithm 6. Interestingly, although for $k = 1$, the Algorithm 6 is identical to 3, for $k > 1$, the algorithms are quite different. The regret guarantee for the **FTRL** policy (41) for the **Generalized *k*-sets** problem follows immediately from the standard results on the regret bound for the **FTRL** policy for general **OCO** problems. The simplified regret bound is given in the following theorem.

Theorem 6 (Regret Bound) *With the learning rate $\eta > 0$, the **FTRL** policy for the generalized **k-sets** problem with the entropic regularizer ensures that*

$$\text{Regret}_T \le \frac{k \ln N/k}{\eta} + 2\eta \sum_{t=1}^T ||\nabla_t^2||_{k,\infty},$$

Algorithm 6 FTRL for the generalized k-**sets** problem with the entropic regularizer

Input: $\boldsymbol{R} \leftarrow \boldsymbol{0}$, learning rate $\eta > 0$

1: **for** every time step t: **do**
2: $\quad \boldsymbol{R} \leftarrow \boldsymbol{R} + \nabla_{t-1}$.
3: $\quad$ Sort the components of the vector $\boldsymbol{R}$ in non-increasing order. Let $R_{(j)}$ denote the j^{th} component of the sorted vector $j \in [N]$.
4: $\quad$ Find the largest index $i^* \in [N]$ such that $(k - i^*)\exp(\eta R_{(i^*)}) \geq \sum_{j=i^*+1}^{N} \exp(\eta R_{(j)})$.
5: $\quad$ Compute the marginal inclusion probabilities as $p_i = \min(1, K\exp(\eta R_i))$, where $K \equiv \frac{k-i^*}{\sum_{j=i^*+1}^{N} \exp(\eta R_{(j)})}$.
6: $\quad$ Using Algorithm 2, sample a $\boldsymbol{k}$-set with the marginal inclusion probabilities $\boldsymbol{p}$.
7: **end for**

where $||\nabla^2_{t,i}||_{k,\infty}$ denotes the sum of the k largest components of the vector ∇^2_t, which is obtained by squaring the vector ∇_t component wise.

Proof: Recall the following general regret bound for the **FTRL** policy from Theorem 5.2 of Hazan (2019). For a bounded, convex and closed set Ω and a strongly convex regularization function $g : \Omega \to \mathbb{R}$, consider the standard **FTRL** updates, *i.e.*,

$$\boldsymbol{x}_{t+1} = \arg\max_{\boldsymbol{x} \in \Omega} \left[\Big(\sum_{s=1}^{t} \nabla_s^T \Big) \boldsymbol{x} - \frac{1}{\eta} g(x), \right] \tag{42}$$

where $\nabla_s = \nabla f_t(\boldsymbol{x}_s), \forall s$. Then, as shown in Hazan (2019), the regret of the **FTRL** policy can be bounded as follows:

$$\text{Regret}_T^{\text{FTRL}} \leq 2\eta \sum_{t=1}^{T} ||\nabla_t||^2_{*,t} + \frac{g(\boldsymbol{u}) - g(\boldsymbol{x}_1)}{\eta}, \tag{43}$$

where the quantity $||\nabla_t||^2_{*,t}$ denotes the square of the dual norm of of the vector induced by the Hessian of the regularizer evaluated at some point $\boldsymbol{x}_{t+1/2}$ lying in the line segment connecting the points $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$. In the **Generalized** k**-set** problem, the Hessian of the entropic regularizer is given by the following diagonal matrix

$$\nabla^2 g(\boldsymbol{p}_{t+1/2}) = \text{diag}([p_1^{-1}, p_2^{-1}, \ldots, p_N^{-1}]).$$

For a vector v, let $||v||_{k,\infty}$ denote the sum of its k largest components. With this notation, we can write

$$||\nabla_t||^2_{*,t} = \sum_{i=1}^{N} p_i \nabla^2_{t,i} \leq ||\nabla^2_t||_{k,\infty},$$

where we have used the fact that $0 \leq p_i \leq 1$ and $\sum_i p_i = k$. In the above, the vector ∇^2_t is obtained by squaring each of the components of the vector ∇_t.

To bound the second term in (43), define a probability distribution $\tilde{\boldsymbol{p}} = \boldsymbol{p}/k$. We have

$$\begin{aligned} 0 \geq g(\boldsymbol{p}) = \sum_i p_i \log p_i &= -k \sum_i \tilde{p}_i \log \frac{1}{p_i} \\ &\overset{\text{(Jensen's inequality)}}{\geq} -k \log \sum_i \frac{\tilde{p}_i}{p_i} = -k \log \frac{N}{k}. \end{aligned}$$

Hence, the regret bound in (43) can be simplified as follows:

$$\text{Regret}_T^{\text{k-set}} \leq \frac{k}{\eta} \log \frac{N}{k} + 2\eta \sum_{t=1}^{T} ||\nabla^2_t||_{k,\infty}.$$

□

11.3.1 Derivation of Algorithm 6

Recall that, via Pinsker's inequality (Fedotov et al., 2003), the entropic regularizer is strongly concave with respect to the ℓ_1 norm. Thus, strong duality holds and the optimal solution to the problem (41) can be obtained by using the KKT conditions (Boyd and Vandenberghe, 2004). To simplify the notations, denote the cumulative sum of the gradient vectors $\sum_{s=1}^{t-1} \nabla_s$ by the vector $\boldsymbol{R}_{t-1}$. Thus, the problem (41) may be explicitly rewritten as follows:

$$\max \sum_{i=1}^{N} p_i R_{t-1,i} - \frac{1}{\eta} \sum_{i=1}^{N} p_i \ln p_i$$

subject to,

$$\sum_{i=1}^{N} p_i = k \tag{44}$$

$$p_i \leq 1, \quad \forall i \tag{45}$$

$$p_i \geq 0, \quad \forall i. \tag{46}$$

By associating the real variable λ with the constraint (44) and the non-negative dual variable ν_i with the i^{th} constraint in (45), we construct the following Lagrangian function:

$$L(\boldsymbol{p}, \lambda, \boldsymbol{\nu}) = \sum_i \left(p_i R_{t-1,i} - \frac{1}{\eta} p_i \ln p_i - \lambda p_i - \nu_i p_i \right) \tag{47}$$

For a set of dual variables $(\lambda, \boldsymbol{\nu})$, we set the gradient of L w.r.t. the primal variables $\boldsymbol{p}$ to zero to obtain:

$$\begin{aligned} p_i &= \exp(\eta R_{t-1,i}) \exp(\lambda \eta - \eta \nu_i - 1) \\ &= K \exp(\eta R_{t-1,i}) \zeta_i, \end{aligned}$$

where $K \equiv \exp(\lambda \eta - 1) \geq 0$ and $\zeta_i \equiv \exp(-\eta \nu_i) \leq 1$. Let us fix the constant K. To ensure the complementary slackness condition corresponding to the constraint (45), we choose the dual variable $\nu_i \geq 0$ such that $p_i = \min(1, K \exp(\eta R_{t-1,i})), \forall i$. Finally, we determine the constant K from the equality constraint (44):

$$\sum_{i=1}^{N} \min(1, K \exp(\eta R_{t-1,i})) = k. \tag{48}$$

For any $k < N$, we now argue that the equation (48) has a unique solution for $K > 0$. The LHS of the equation (48) is a continuous, non-decreasing function of K and takes value in the interval $[0, N]$. Hence, by the intermediate value theorem, the equation (48) has at least one solution. Furthermore, at the equality, at least one of the constituent terms will be strictly smaller than one. Since this term is strictly increasing with K, the proposition follows.

To efficiently solve the equation (48), we sort the cumulative request vector $\boldsymbol{R}_{t-1}$ in non-increasing order. Let $R_{t-1,(i)}$ denote the i^{th} term of the sorted vector. Let i^* be the largest index for which $K \exp(\eta R_{t-1,(i^*)}) \geq 1$. Then, the equation (48) can be written as:

$$i^* + K \sum_{j=i^*+1}^{N} \exp(\eta R_{t-1,(j)}) = k.$$

i.e.,

$$K = \frac{k - i^*}{\sum_{j=i^*+1}^{N} \exp(\eta R_{t-1,(j)})}. \tag{49}$$

where i^* is the largest index to satisfy the following constraint:

$$(k - i^*) \exp(\eta R_{t-1,(i^*)}) \geq \sum_{j=i^*+1}^{N} \exp(\eta R_{t-1,(j)}). \tag{50}$$

Hence, the optimal index i^* may be determined in linear time by starting with $i^* = N$ and decreasing the index i^* by one until the condition (50) is satisfied. Once the optimal i^* is found, the optimal value of the constant K may be obtained from equation (49). The overall complexity of the procedure is dominated by the sorting step and is equal to $O(N \ln N)$. However, since only one index changes at a time, in practice, the average computational cost is much less.

11.4 Approximating arbitrary Set functions by Modular functions

For completeness, here we outline the main steps involved for proving Theorem 3 (see also Wu et al. (2019) for an exposition).

Theorem 7 (Sandwich Theorem, Iyer and Bilmes (2012)) *For a given set X and any set function $f : 2^X \to \mathbb{R}$ and any set $Y \subseteq X$, there are two modular functions $m_u : 2^X \to \mathbb{R}$ and $m_l : 2^X \to \mathbb{R}$ such that $m_l \leq f \leq m_u$ and $m_l(Y) = f(Y) = m_u(Y)$. Furthermore, the functions m_u and m_l can be explicitly expressed in terms of the function f.*

The above Sandwich theorem is a consequence of a series of results that we briefly describe below. Recall that a set function $f : 2^X \to \mathbb{R}$ is called submodular if for all $A, B \subseteq 2^X$, we have

$$f(A \cup B) + f(A \cap B) \leq f(A) + f(B).$$

The following two lemmas approximate any submodular function by modular functions. For any two sets $A, B \subset 2^X$, define $f(A|B) := f(A \cup B) - f(B)$.

Lemma 1 (Upper bound (Iyer and Bilmes, 2012)) *For any submodular function $f : 2^X \to \mathbb{R}$, and $Y \subseteq X$, there exists a modular function $m_u(A)$ such that $m_u \geq f$ and $m_u(Y) = f(Y)$. One such candidate modular function m_u is given as follows:*

$$m_u(A) = f(Y) + \sum_{j \in A \setminus Y} f(j|\emptyset) - \sum_{j \in Y \setminus A} f(j|Y \setminus j). \tag{51}$$

Lemma 2 (Lower bound (Iyer and Bilmes, 2012)) *For any submodular function $f : 2^X \to \mathbb{R}$, and $Y \subseteq X$, there exists a modular function $m_l(A)$ such that $m_l \leq f$ and $m_l(Y) = f(Y)$. One such candidate modular function m_l is given as follows:*
Define any permutation (ordering) of the elements of $X = \{x_1, x_2, \ldots, x_{|X|}\}$. Subsequently define $Y = \{x_1, x_2, \ldots, x_{|Y|}\}$ and sets $S_i = \{x_1, x_2, \ldots, x_i\}$. Define $m_l(\emptyset) = f(\emptyset)$. Then, for $\emptyset \neq A \subseteq X$,

$$m_l(A) = m_l(\emptyset) + \sum_{x_i \in A} (f(S_i) - f(S_{i-1})). \tag{52}$$

Finally, the following result shows that any arbitrary set function can be expressed as the difference of two submodular functions.

Lemma 3 (Difference of Submodular functions (Narasimhan and Bilmes, 2012)) *Every set function $f : 2^X \to \mathbb{R}$ can be expressed as the difference of two monotone nondecreasing submodular functions g and h, i.e., $f = g - h$.*

Iyer and Bilmes (2012) gives an exact characterization of the functions g and h as follows: let h be any strictly submodular function. Compute

$$\beta = \min_{Y \subset Z \subseteq X \setminus j} \Big(h(j|Y) - h(j|Z) \Big). \tag{53}$$

For example, by taking $h(Y) := \sqrt{|Y|}$, we have $\beta = 2\sqrt{N-1} - \sqrt{N} - \sqrt{N-2} = O(\frac{1}{N^{3/2}})$, where $N = |X|$. Similarly, define

$$\alpha(f) = \min_{Y \subset Z \subseteq X \setminus j} \Big(f(j|Y) - f(j|Z) \Big). \tag{54}$$

By definition $\alpha \geq 0 \iff f$ is submodular. In that case, we can take $g = f, h = 0$ and we are done. In case $\alpha < 0$, consider any $\alpha' \leq \alpha$. Then, Iyer and Bilmes (2012) showed that the function f can be expressed as $f = \hat{g} - \hat{h}$ where

$$\hat{g} = f + \frac{|\alpha'|}{\beta} h, \text{ and } \hat{h} = \frac{|\alpha'|}{\beta} h, \tag{55}$$

where $\hat{g}$ and $\hat{h}$ can be easily seen to be submodular.

Note that computing the parameter $\alpha(f)$ for any arbitrary set function f could be intractable (Iyer and Bilmes, 2012). However, we can readily obtain a lower bound α' to α for monotone reward functions. We have

$$\begin{aligned}
\alpha &= \min_{Y \subset Z \subseteq X \setminus j} f(j|Y) - f(j|Z) \\
&\geq \min_{Y \subseteq X} f(j|Y) - \max_{Z \subseteq X} f(j|Z) \\
&\overset{(a)}{\geq} - \max_{Z \subseteq X} f(j|Z) =: \alpha',
\end{aligned}$$

where the inequality (a) follows from the monotonicity of the function f. In other words, $|\alpha'|$ is largest marginal gain of adding an element to any set $Z \subseteq X$ for the function f. To proceed further, as stated in Eqn. (15), we assume the function f to be smooth, *i.e.,* $\forall S \subseteq [N], x \in S$, we have:

$$|f(S) - f(S \setminus \{x\})| \leq G, \tag{56}$$

for some finite constant G. Under the smoothness assumption, we can set $|\alpha'| = G$. Combining Lemma 1, Lemma 2, and Lemma 3, we can now explicitly write down the expressions for the modular functions m_l and m_u appearing in Theorem 3 as follows:

$$m_l = m_l^g - m_u^h \tag{57}$$

$$m_u = m_u^g - m_l^h. \tag{58}$$

We now proceed to derive an explicit expression for the modular function m_l.

Expression for the function m_l: For a given ordering π of the elements of X, let $\sigma(i) \equiv \pi^{-1}(i)$ denote the position of the element i in the ordering. Setting $Y = [N]$ and choosing $h(S) = \sqrt{|S|}$ in Lemma 1 and Lemma 2, we have the following expression for the function m_l:

$$\begin{aligned}
m_l^{\hat{g}}(i) &= \hat{g}(S_{\sigma(i)}) - \hat{g}(S_{\sigma(i)-1}) \\
&= f(S_{\sigma(i)}) - f(S_{\sigma(i)-1}) + \frac{|\alpha'|}{\beta} \left(h(S_{\sigma(i)}) - h(S_{\sigma(i)-1}) \right) \\
&= f(S_{\sigma(i)}) - f(S_{\sigma(i)-1}) + \frac{|\alpha'|}{\beta} \left(\sqrt{|S_{\sigma(i)}|} - \sqrt{|S_{\sigma(i)-1}|} \right)
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
m_u^{\hat{h}}(i) &= \frac{|\alpha'|}{\beta} \left(h([N]) + \sum_{j \in i \setminus [N]} h(j|\emptyset) - \sum_{j \in [N] \setminus i} h(j|[N] \setminus j) \right) \\
&= \frac{|\alpha'|}{\beta} \left(h([N]) - \sum_{j \in [N] \setminus i} h(j|[N] \setminus j) \right) \\
&= \frac{|\alpha'|}{\beta} \left(\sqrt{N} - (N-1)(\sqrt{N} - \sqrt{N-1}) \right) =: C
\end{aligned} \tag{59}$$

Hence, the i^{th} component of the function m_l is given by:

$$\begin{aligned}
g(i) &\equiv m_l(i) \\
&= m_l^{\hat{g}}(i) - m_u^{\hat{h}}(i) \\
&= f(S_{\sigma(i)}) - f(S_{\sigma(i)-1}) + \frac{|\alpha'|}{\beta} \left(\sqrt{|S_{\sigma(i)}|} - \sqrt{|S_{\sigma(i)-1}|} \right) - C.
\end{aligned} \tag{60}$$

Eqn. (60) gives an explicit and efficiently computable expression for the lower modular function m_l which we use in our online learning policy.

Define a "centered" gradient vector $\tilde{g}$ corresponding to the vector g as $\tilde{g}(i) = g(i) + C, \forall i \in [N]$, where the constant C is defined in Eqn. (59). Now for any feasible inclusion probability vector p, we have

$$\langle \tilde{g}(i), p \rangle = \langle g, p \rangle + Ck,$$

where we have used the feasibility constraint (2). Hence, the online policy and the regret as defined in Eqn.(17) remain unchanged if we replace the vectors $\{g_t\}_{t\geq 1}$ with their centered counterparts $\{\tilde{g}_t\}_{t\geq 1}$.

Using triangle inequality, we can upper bound the magnitude of each component of the centered vector $\tilde{g}$ as:

$$\begin{aligned} |\tilde{g}(i)| &\leq |f(S_{\sigma(i)}) - f(S_{\sigma(i)-1})| + \frac{|\alpha'|}{\beta}\left|\sqrt{|S_{\sigma(i)}|} - \sqrt{|S_{\sigma(i)-1}|}\right| \\ &\overset{(a)}{\leq} G + \frac{G}{\beta} = O(GN^{3/2}) \end{aligned} \tag{61}$$

where (a) holds because by assumption f is smooth with parameter $G, \beta \sim O(1/N^{3/2})$ for the particular choice of the h function above, and $|\alpha'| \leq G$. Hence, the sum of the largest k components of the vector $(\tilde{g}_1^2, \tilde{g}_2^2, \ldots, \tilde{g}_N^2)$ can be bounded by:

$$||\tilde{\boldsymbol{g}}^2||_{k,\infty} \leq O((\sqrt{k}GN^{3/2})^2).$$

Note that the upper bound in Eq. (61) holds for any monotone set function f. As shown below, the above bound can be improved in the special case when the set function f is known to be submodular.

Special Case - Submodular f : As discussed above, if the function f is restricted to be submodular, we can directly use Lemma 2 to obtain an expression for the modular function m_l as follows: Fix any permutation of the elements of $[N]$.

$$g(i) = m_l(i) = f(S_{\sigma(i)}) - f(S_{\sigma(i)-1}).$$

This gives the following bound $|g(i)| \leq G, \forall i \in [N]$. Hence, proceeding as above, we have:

$$||\boldsymbol{g}^2||_{k,\infty} \leq O((\sqrt{k}G)^2).$$

11.5 Proof of Theorem 5

Outline: We seek to obtain a tight lower bound to the regret of the k-experts problem with the Max-reward variant. Before we delve into the technical details, we first outline the main steps behind the proof. We define an i.i.d. reward structure where the reward of any expert at each slot is distributed as i.i.d. Bernoulli with parameter $p = 1/2k$. Next, we compute a lower bound to the expected cumulative reward accrued by the static offline oracle policy by constructing a set S^* consisting of k experts, as outlined next. First, we divide the set of N experts into k disjoint partitions, each consisting of $\frac{N}{k}$ experts [3]. Denote the set of experts in the i^{th} partition by $P_i, 1 \leq i \leq k$. Let $e_i^* \in P_i$ be the expert from the i^{th} having the highest cumulative reward up to time T in hindsight. Finally, we define the set $S^* \equiv \{e_i^*, 1 \leq i \leq k\}$. Trivially, the cumulative reward accrued by the optimal offline oracle is lower bounded the reward accrued by the set of experts in S^*. Furthermore, since the experts $e_i^*, 1 \leq i \leq k$ are identically distributed and independent of each other, the computation of the reward accrued by the set S^* becomes tractable. In the following, we show that the expected reward accumulated by the set S^* is given by the expectation of the maximum of k i.i.d. Binomial random variables. The regret lower bound in Theorem 5 finally follows from a tight non-asymptotic lower bound to this expectation, which we believe, has not appeared in this form before.

[3] For ease of typing, we assume that the number of experts N is divisible by k. If that is not the case, consider the first $\tilde{N} = k\lfloor \frac{N}{k} \rfloor$ experts only.

Proof: We use the standard "randomization trick" to obtain a lower bound to the worst-case regret:

$$\max_{\{\boldsymbol{r}_t\}_{t=1}^T} \mathcal{R}_T \geq \mathbb{E}_r\big(\mathcal{R}_T\big), \tag{62}$$

where we use the symbol $\mathbb{E}_r$ to convey that the expectation is taken over a random binary input reward sequence $\{r_{t,i}\}_{i\in[N],1\leq t\leq T}$, where the random rewards $r_{t,i}$'s are taken to be i.i.d. $\sim \text{Bern}(p)$, for some parameter $p \in [0,1]$, that will be fixed later. Using the definition of the regret in Eq. (1), we obtain:

$$\max_{\{\boldsymbol{r}_t\}_{t=1}^T} \mathcal{R}_T \geq \texttt{OPT} - \sum_{t=1}^T \mathbb{E}_r\big(\max_{i\in S_t} r_{t,i}\big), \tag{63}$$

where we denote

$$\texttt{OPT} = \mathbb{E}_r\big(\max_{S\subset[N]:|S|=k} \sum_{t=1}^T \max_{i\in S} r_{t,i}\big). \tag{64}$$

Since the rewards $r_{t,i},\ i\in[N]$ are i.i.d.$\sim \text{Bern}(p)$, for any choice of the set S_t, we have:

$$\mathbb{E}_r\big(\max_{i\in S_t} r_{t,i}\big) = \mathbb{P}\big(\max_{i\in S_t} r_{t,i} = 1\big) = 1-(1-p)^k. \tag{65}$$

It now remains to establish a lower bound to the quantity $\texttt{OPT}$. In order to do that, we first make the trivial observation that, for any subset $S\subseteq[N]$ with cardinality k, the following holds true:

$$\texttt{OPT} \geq \sum_{t=1}^T \mathbb{E}\big(\max_{i\in S} r_{t,i}\big). \tag{66}$$

Note that in the above, we can allow random S, that might depend on the particular realizations of the random reward sequence. Using this observation, we now use the bound (66) with the set $S^\star$ as defined below: Divide the set N experts into k disjoint partitions $B_1,\cdots,B_k$, each of size $b=N/k$, such that

$$B_l = \{(l-1)b+1,\cdots,lb\},\quad 1\leq l\leq k. \tag{67}$$

Finally, we construct the set $S^\star \equiv \{i_1,\cdots,i_k\}$, where, $i_l = \arg\max_{j\in B_l} X_{T,j},\ 1\leq l\leq k$, where $X_{T,j}=\sum_{t=1}^T r_{t,j}$. In other words, i_l is the (random) index of the expert in the l^{th} partition such that it has the highest cumulative reward in hindsight. By construction, the random indices $i_1,\cdots i_k$ are independent of each other. Hence, the random rewards $r_{t,i},\ i\in S^\star$ are independent Bernoulli random variables with some parameter q, that we will determine shortly. Using the observation that for a fixed $1\leq l\leq k$, the random variables r_{t,i_l} for $t=1,\cdots,T$, are identically distributed, it follows that $\mathbb{E}(r_{t,i_l})$ is identical for all t for a fixed l, so that

$$q \equiv \mathbb{E}(r_{t,i_l}) = \frac{1}{T}\mathbb{E}(X_{T,i_l}) = \frac{1}{T}\mathbb{E}(\max_{j\in B_l} X_{T,j}). \tag{68}$$

Hence, using the lower bound (66), we have

$$\texttt{OPT} \geq \sum_{t=1}^T \big(1-(1-q)^k\big) = T(1-(1-q)^k). \tag{69}$$

Hence, combining Eqns. (63), (65) with the lower bound in Eqn. (69), we have the following regret lower bound in terms of the yet undetermined parameter q:

$$\max_{\{\boldsymbol{r}_t\}_{t=1}^T} \mathcal{R}_T \geq T\left((1-p)^k-(1-q)^k\right). \tag{70}$$

Since the function $(1-p)^k$ is convex in p, linearizing the function around the point q yields the following lower bound for regret:

$$\max_{\{\boldsymbol{r}_t\}_{t=1}^T} \mathcal{R}_T \geq kT(q-p)(1-q)^{k-1}. \tag{71}$$

To proceed further, we need to estimate q by finding tight upper and lower bounds for it.

1. Upper bounding q: Since the random variables $X_{T,j},\ j \in B_1$ are i.i.d. Binomial, and hence subGaussian with mean $\mu = \mathbb{E}X_{T,1} = Tp$ and variance $\sigma^2 = Tp(1-p)$, it follows from Massart's maximal lemma for Gaussians (Massart, 2007) that:

$$\begin{aligned} q - p &= \frac{1}{T}\Big(\mathbb{E}(\max_{j\in B_1} X_{T,j}) - pT\Big) \\ &\leq \sqrt{\frac{2p(1-p)\ln(N/k)}{T}}. \end{aligned}$$

In particular, for a large enough horizon-length $T \geq 8(\frac{1}{p}-1)\ln(\frac{N}{k})$, from the above we have the following upper bound for q:

$$q \leq \frac{3p}{2}. \tag{72}$$

2. Lower bounding q: We have

$$\begin{aligned} q - p &= \frac{1}{T}\mathbb{E}\Big(\max_{j\in B_1}(X_{T,j} - Tp)\Big) \\ &= \frac{1}{T}\mathbb{E}\Big(\max_{j\in B_1}(X_{T,j} - Tp)\mathbb{1}\big(\max_{j\in B_1} X_{T,j} < Tp\big)\Big) \\ &+ \frac{1}{T}\mathbb{E}\Big(\max_{j\in B_1}(X_{T,j} - Tp)\mathbb{1}\big(\max_{j\in B_1} X_{T,j} \geq Tp\big)\Big) \\ &\overset{\text{(def.)}}{=} \frac{I_1 + I_2}{T}. \end{aligned} \tag{73}$$

Now, we separately lower bound each of the quantities I_1 and I_2 as defined above.

2.1. Lower bounding I_1: We have the following inequalities:

$$\begin{aligned} I_1 &\equiv \mathbb{E}\Big(\max_{j\in B_1}(X_{T,j} - Tp)\mathbb{1}\big(\max_{j\in B_1} X_{T,j} < Tp\big)\Big) \\ &\overset{(a)}{\geq} \max_{j\in B_1}\mathbb{E}\Big((X_{T,j} - Tp)\mathbb{1}(X_{T,j} < Tp)\prod_{i\in B_1, i\neq j}\mathbb{1}(X_{T,i} < Tp)\Big) \\ &\overset{(b)}{=} \mathbb{E}\Big((X_{T,1} - Tp)\mathbb{1}(X_{T,1} < Tp)\Big)\Big(\mathbb{P}(X_{T,1} < Tp)\Big)^{b-1} \\ &\overset{(c)}{\geq} -\mathbb{E}\big|X_{T,1} - Tp\big|\Big(\mathbb{P}(X_{T,1} < Tp)\Big)^{b-1} \\ &\overset{(d)}{\geq} -\sqrt{Tp(1-p)}\Big(\mathbb{P}(X_{T,1} < Tp)\Big)^{b-1} \\ &\overset{(e)}{\geq} -(\frac{3}{4})^{b-1}\sqrt{Tp(1-p)}. \end{aligned} \tag{74}$$

in the above,

1. inequality (a) follows from Jensen's inequality and the trivial fact that $\mathbb{1}(\max_{j\in B_1} X_{T,j} < Tp) = \mathbb{1}(X_{T,j} < Tp)\prod_{i\in B_1, i\neq j}\mathbb{1}(X_{T,i} < Tp)$
2. inequality (b) follows from the fact that the collection of r.v.s $\{X_{T,j}, j \in B_1\}$ are independent and identically distributed
3. inequality (c) follows from the fact that $(X_{T,1} - Tp)\mathbb{1}(X_{T,1} < Tp) \geq -\big|X_{T,1} - Tp\big|$,
4. in inequality (d), we have used Jensen's inequality with the fact that $X_{T,1} \sim \text{Binomial}(T, p)$
5. finally, in inequality (e), we have used Theorem 1 from Greenberg and Mohri (2014) which states that for $p > 1/T$ we have $\mathbb{P}(X_{T,1} \geq Tp) \geq 1/4$.

2.2. Lower bounding I_2: Using Markov's inequality, we have for any $s \geq 0$:

$$\begin{aligned} I_2 &\geq s\mathbb{P}\big(\max_{j\in B_1} X_{T,j} > s + Tp\big) \\ &\overset{(a)}{=} s\left(1 - \big(\mathbb{P}\big(X_{T,1} \leq s + Tp\big)\big)^b\right) \\ &\overset{(b)}{\geq} s\left(1 - \big(\Phi(\frac{s}{\sqrt{Tp(1-p)}})\big)^b\right). \end{aligned} \tag{75}$$

where in step (a), we have used the independence of the r.v.s $X_{T,j}, j \in B_1$ and in step (b), we have used Slud's inequality (Cesa-Bianchi and Lugosi, 2006). Note that in the above, we use the standard notation where $\Phi(\cdot)$ denotes the CDF of the standard Normal variable.

Observe that for any $u > 0$, we can upper bound the normal CDF as:

$$\begin{aligned} \Phi(u) &= 1 - \frac{1}{\sqrt{2\pi}}\int_u^\infty e^{-x^2/2}dx \\ &\leq 1 - \frac{1}{\sqrt{2\pi}}\int_u^{2u} e^{-x^2/2}dx \\ &\leq 1 - \frac{ue^{-2u^2}}{\sqrt{2\pi}}. \end{aligned} \tag{76}$$

By making a change of variable $u \leftarrow \frac{s}{\sqrt{Tp(1-p)}}$ in Eqn. (75), the quantity I_2 can be lower bounded as:

$$I_2 \geq \sqrt{Tp(1-p)}\left[u\left(1 - \big(1 - \frac{ue^{-2u^2}}{\sqrt{2\pi}}\big)^b\right)\right]. \tag{77}$$

Choosing $u = \sqrt{\frac{\ln b}{2}}$ and using the standard inequality $1 - x \leq e^{-x}, \forall x$, from the above we have:

$$I_2 \geq c_1\sqrt{Tp(1-p)\ln b}, \tag{78}$$

where $c_1 \equiv \frac{1}{\sqrt{2}}(1 - e^{-\sqrt{\ln b/4\pi}})$.

Combining the bounds for I_1 and I_2 from (74) and (78), we obtain the following lower bound for q from Eqn. (73) valid for $b \equiv \frac{N}{k} \geq 7$:

$$q - p \geq \frac{c_2}{T}\sqrt{Tp(1-p)\ln\frac{N}{k}}, \tag{79}$$

where $c_2 \geq 0.1$ is an absolute constant.

3. Lower bounding the regret: Finally, we choose $p = \frac{1}{2k}$. Substituting the bounds (72) and (79) into the regret lower bound (71), for $T \geq 16k\ln(\frac{N}{k})$ and $\frac{N}{k} \geq 7$, we obtain:

$$\max_{\{\boldsymbol{r}_t\}_{t=1}^T} \mathcal{R}_T \geq c_2 k\sqrt{\frac{T}{2k}(1 - \frac{1}{2k})\ln\frac{N}{k}}\left(1 - \frac{3}{4k}\right)^{k-1} \geq c_3\sqrt{kT\ln\frac{N}{k}}, \tag{80}$$

where $c_3 \geq 0.02$ is an absolute constant. □

12 Additional Experimental Results

In Figure 4, we plot the hit rates (*i.e.,* the fraction of correct predictions) of various prediction policies for the `k-sets` problem for the MovieLens dataset. From the plots, we observe that by selecting only 30% of the elements (*i.e.,* $k/N = 0.3$), the `SAGE` policy with $\pi_{\text{base}} =$ `Hedge` achieves a hit rate of at least 60%. We also measure the performance of the proposed policy for the `k-sets` problem on Wiki-CDN dataset (Berger et al., 2018). This dataset contains publicly available Wikipedia CDN request traces from a server located in San Francisco. It contains trace for $T \sim 10^5$ time stamps and $N \sim 2500$ files. We compare the performance of different policies in terms of the normalized regret and hit rates in Figure 5 and 6 respectively. From the plots, we observe that the `SAGE` policy outperforms other benchmarks by a large margin.

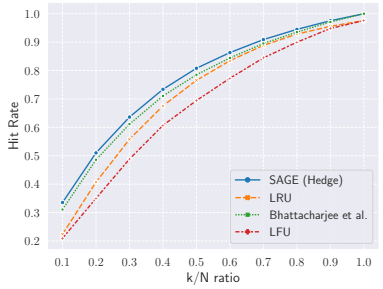

Figure 4: Comparison among different prediction policies in terms of hit rates (fraction of correct predictions) for different values of $k/N, N \sim 2400$ for the MovieLens dataset.

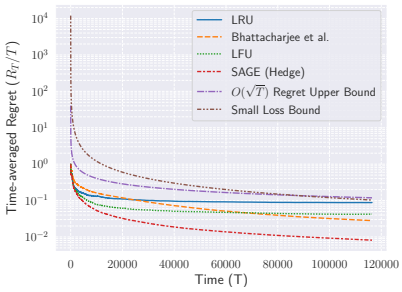

Figure 5: Comparison among different prediction policies in terms of normalized regret $\frac{R_T}{T}$ with $k/N = 0.1, N \sim 2500$ for the Wiki-CDN dataset.

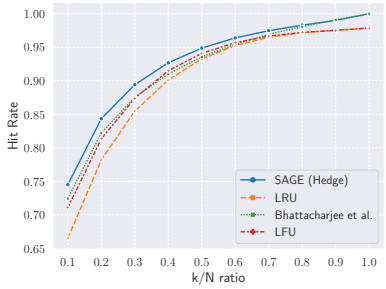

Figure 6: Comparison among different prediction policies in terms of hit rates (fraction of correct predictions) for different values of $k/N, N \sim 2500$ for the Wiki-CDN dataset.

12.1 Experiments with the learning policy for Monotone Reward Functions (Section 6)

In our experiments for the general monotone reward functions, we use a subset of the MovieLens dataset with $T \sim 200$ and $N = 100$. Similar to Section 8, we assume that the movies are sorted according to genres so that if movie i is chosen by the user at round t, then the reward vector, $\boldsymbol{r}_t \in [0,1]^N$, is given as $r_{t,j} = 1 - \frac{1}{N}|j - i|$. For a reward vector $\boldsymbol{r}_t$ and real-valued function $v : \mathbb{R}^N \to \mathbb{R}_{\geq 0}$, we define a monotone set function $f_t : 2^{[N]} \to \mathbb{R}_{\geq 0}$ as $f_t(S) = v(\boldsymbol{r}_t(S)), \forall S \subseteq [N]$ where $[r_t(S)]_i = r_{t,i} \cdot \mathbb{1}(i \in S)$. According to the k-experts setting, we assume that the learner receives a reward of $f_t(S_t)$ for predicting the set S_t. In our experiments, we consider two different reward functions $v : \boldsymbol{x} \mapsto ||\boldsymbol{x}||_p$, with $p = 2$ and $p = \infty$. In Figure 7 and 8, we plot the rewards obtained by the learner as a fraction of the total possible rewards (when all the elements are selected). From the plots, it is clear that the proposed policy has excellent performance for both reward functions, as it achieves a large fraction of the total possible reward by using only a small fraction of the experts.

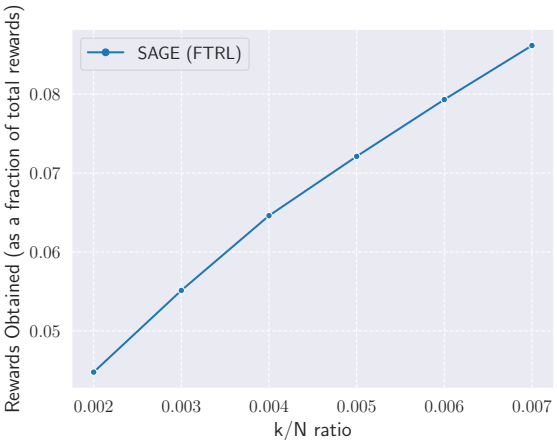

Figure 7: Performance of prediction policy in terms of fraction of total possible reward (by selecting all the elements) obtained for $N = 1000, v : \boldsymbol{x} \mapsto ||\boldsymbol{x}||_2$ for the MovieLens dataset.

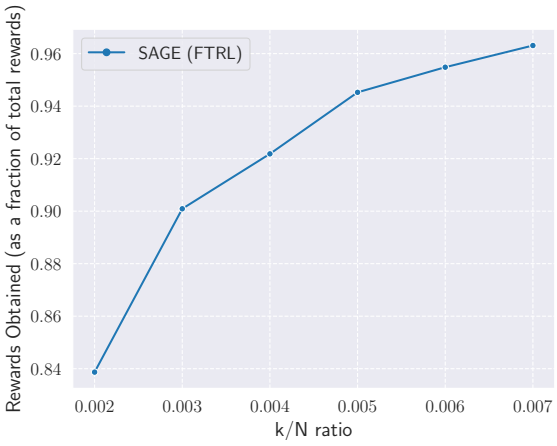

Figure 8: Performance of prediction policy in terms of fraction of total possible reward (by selecting all the elements) obtained for $N = 1000, v : \boldsymbol{x} \mapsto ||\boldsymbol{x}||_\infty$ for the MovieLens dataset.