k-experts - Online Policies and Fundamental Limits

Samrat Mukhopadhyay[1], Sourav Sahoo[2], and Abhishek Sinha[*2]

[1]Department of Electronics Engineering, IIT (ISM) Dhanbad
[2]Department of Electrical Engineering, IIT Madras

October 18, 2021

Abstract

This paper introduces and studies the k-`experts` problem - a generalization of the classic *Prediction with Expert's Advice* (*i.e.*, the `Experts`) problem. Unlike the `Experts` problem, where the learner chooses exactly one expert, in this problem, the learner selects a subset of k experts from a pool of N experts at each round. The reward obtained by the learner at any round depends on the rewards of the selected experts. The k-`experts` problem arises in many practical settings, including online ad placements, personalized news recommendations, and paging. Our primary goal is to design an online learning policy having 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. We show that, for many related problems, `SAGE` improves upon the state-of-the-art bounds for regret and computational complexity. Furthermore, going beyond the notion of regret, we characterize the mistake bounds achievable by online learning policies for a class of 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 and Related Work

The classic *Prediction with Expert's Advice* problem, also known as the `Experts` problem, 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. Assume that N experts are indexed by the set $[N] = \{1, 2, \ldots, N\}$. At 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 current 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 policy that incurs a small *regret*. Recall that the regret of an online policy over a time horizon of length T is defined as the difference between the reward accumulated by the best fixed expert in hindsight and the total expected reward accrued by the online policy over the time horizon (viz. Eqn. (1)). Many online policies achieving sublinear regrets in this problem are known, most notably, `Hedge` (Vovk, 1998; Freund and Schapire, 1997).

In this paper, we introduce and study the k-`experts` problem - a generalization of the `Experts` problem. In the k-`experts` problem, instead of selecting only one expert at each round, the learner selects a subset $S_t \subseteq [N]$ consisting of k experts at round t $(1 \leq k \leq N)$. The reward $q(S_t, \boldsymbol{r}_t)$, received by the learner at round t, depends on the rewards of the experts in the chosen set S_t. Table 1 describes some variants of the k-`experts` problem obtained by choosing different functional forms for the reward function $q(\cdot)$. 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 marginal (conditional) probability that the i^{th} expert is included in the set S_t, given the history of the game. Then, we can express the (conditional) expected reward for the t^{th} round as $\mathbb{E}q^{\texttt{Sum-reward}}(S_t, \boldsymbol{r}_t) = \boldsymbol{r}_t \cdot \boldsymbol{p}_t$. The `Max-reward`, ℓ_p-`reward`, and `Pairwise-reward` variants

*Correspondence to: `abhishek.sinha@ee.iitm.ac.in`

Table 1: Variants of the $\boldsymbol{k}$**-experts** problem

`Sum-reward`	`Max-reward`	ℓ_p`-reward` $(p \geq 1)$	`Pairwise-reward`
$q_{\texttt{sum}} = \sum_{i \in S_t} r_{ti}$	$q_{\texttt{max}} = \max_{i \in S_t} r_{ti}$	$q_{\ell_p} = \big(\sum_{i \in S_t} r_{ti}^p\big)^{1/p}$	$q_{\texttt{pair}} = \sum_{i,j \in S_t} r_{it} r_{jt}$

are defined analogously. However, unlike the `Sum-reward` variant, the expected accrued reward for the latter variants depends on higher-order joint inclusion probabilities as well (as opposed to only marginals). For each of the above variants, we formulate the problem of designing an online expert selection policy that minimizes the regret $\mathcal{R}_T$ over a horizon of length T as defined below:

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

In the above definition, the expectation in the second term is taken with respect to any random choice made by the learner.

Related problems: An important special case of the $\boldsymbol{k}$**-experts** problem is the problem of **Online** $\boldsymbol{N}$**-ary prediction with** $\boldsymbol{k}$**-sets** (in brief, the k`-sets` problem) (Koolen et al., 2010). In this problem, a learner sequentially predicts the next symbol in 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 the predicted set S_t contains the symbol y_t or incorrect otherwise. 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 time period. It is easy to see that the above problem is a special case of the $\boldsymbol{k}$`-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 corresponds to 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. Fifty years later, Rakhlin and Sridharan (2016) generalized Cover's result to an arbitrary alphabet of size N, but still requiring $k = 1$. The fundamental limits of the achievable prediction error for the k`-sets` problem with an arbitrary N and k has long remained an 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 `Experts` problem with a 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 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 $O(N^2)$. We refer the readers to Takimoto and Hatano (2013) for a survey of the efficient projection and decomposition procedures for the k`-sets` and other online combinatorial optimization problems. The k`-sets` problem has also been investigated by Daniely and Mansour (2019), where the authors referred to as the *paging* problem. The authors alleviated the complexity of the naïve `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*-based policy. Although simple to implement, their algorithm does not admit a data-dependent small-loss bound.

Our contributions: Our contributions can be summarized as follows:

1. 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 online N-ary prediction problem with k-sets.

2. In Section 4, we present `SAGE` - an efficient, projection-free, regret-optimal online prediction framework. Using `SAGE`, we design an efficient online policy for the k-`sets` problem. The proposed policy runs in linear time, admits a small-loss bound, and breaks the existing quadratic computational barrier (see Table 2). We also design a different OCO-based prediction policy for a generalized version of the k-`sets` problem.

3. In Section 5, we present a `SAGE`-based *improper* learning policy that achieves a $O(\sqrt{T})$ regret for the `Pairwise-reward` version of the k-`experts` problem.

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

5. Finally, in Section 7, we numerically compare the performance of the proposed policies with other benchmarks on some standard datasets.

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

1.1 Key Insight for `SAGE`

As pointed out earlier, the expected `Sum-reward` for any policy depends *only* on the first-order marginal inclusion probabilities and *not* on the higher-order joint inclusion distribution. In particular, any two online prediction policies, that have the same conditional marginal inclusion probabilities, yield *exactly* the same expected sum reward per round. This simple observation leads to the `SAGE` meta-algorithm described below.

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**

Clearly, the above procedure yields the same performance (regret) as the base policy π_{base}. However, the advantage of the above three-step process stems from the fact that, unlike the base policy (which could be computationally intractable), the `SAGE` policy can be efficiently implemented for many problems. We show that when `Hedge` is used as a base policy for the $\boldsymbol{k}$-**sets** problem, the marginalization in line 3 reduces to the evaluation of certain elementary symmetric polynomials, that can be efficiently computed using Fast Fourier Transform (FFT) techniques. Furthermore, an efficient sampler for line 4 can be borrowed from the statistical sampling literature, reviewed in Section 2.

`SAGE` is flexible and can be easily augmented with heuristics when the exact marginalization in line 3 is difficult to obtain. This unlocks the potential application of a host of off-the-shelf algorithms from the Probabilistic Graphical Model literature to online learning problems. However, we would like to remind the reader that `SAGE` policy is not necessarily regret-optimal when the expected reward depends on higher-order inclusion probabilities (such as the k-`experts` problem with `Pairwise-reward` or `Max-reward`), for which we need other techniques (see Section 5).

2 Sampling without Replacement

Our proposed online policy makes critical use of systematic sampling techniques from statistics, which we briefly review in this section. Consider the problem of sampling without replacement where one needs to randomly select a k-set $S \in \binom{[N]}{k}$ such that an item $i \in [N]$ is included in the

Table 2: Performance Comparison among Different Policies

Policies	Reference	Regret bound	Complexity
FTPL (Gaussian perturbation)	Cohen and Hazan (2015)	$2\sqrt{2Tk\ln\binom{N}{k}}$	$\tilde{O}(N)$
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	$2\sqrt{2kT\ln\frac{N}{k}}$	$\tilde{O}(N)$

set S with a pre-specified marginal 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 the following consistency condition must be satisfied for any feasible marginal inclusion probability vector $\boldsymbol{p}$:

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

Surprisingly, 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 in Algorithm 2.

Algorithm 2 Madow's Sampling Scheme

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

Correctness: The correctness of Madow's sampling scheme is easy to establish. Due to the necessary condition (2), it is easy to see 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

We begin with the following prediction problem studied by Cover (1966) - assume that an adversary secretly selects a binary sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$, that is revealed to a learner sequentially. Upon seeing the initial segment of the sequence y_1^{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 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 a binary 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. As shown in Eqn. (5) below, it readily follows that irrespective of the prediction algorithm $\mathcal{A}$, the average value of the fraction of errors $\mu_{\mathcal{A}}(\cdot)$ over all possible 2^T binary sequences is precisely $1/2$. A loss function $\phi : \{\pm 1\}^T \to [0, 1]$ is said to be achievable if there exists an online prediction policy $\mathcal{A}$ such that the average prediction error on any sequence is upper bounded by the function $\phi(\cdot)$, *i.e.,* $\mu_{\mathcal{A}}(\boldsymbol{y}) \leq \phi(\boldsymbol{y}), \forall \boldsymbol{y}$. An immediate problem 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 $\phi : \{\pm 1\}^T \to [0,1]$ to be *stable* if it satisfies the following inequality for all $\boldsymbol{y} \in \{\pm 1\}^T$ and for all $1 \leq t \leq T$:

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

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

Theorem 1 ((Cover, 1966))**.** *Suppose the loss function $\phi : \{\pm 1\}^T \to [0,1]$ is stable. Then $\phi(\cdot)$ is achievable if and only if $\mathbb{E}\phi(\boldsymbol{z}) \geq 1/2$, where the expectation is taken with respect to the i.i.d. uniform distribution over $[N]^T$.*

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

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

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

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

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

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

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

We now state the following theorem that generalizes Cover's result by showing that conditions (6) and (7) 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 eventually achieves the loss function $\phi(\cdot)$. In Phase-II, we sample a k-set $S_t \subseteq [N]$ consistent with the marginal inclusion probabilities $\boldsymbol{p}_t$ using Algorithm 2. Please refer to Section A.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 quickly computed via alternative methods, resulting in low-regret efficient online prediction policies.

4 Efficient Policies for k-sets

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` algorithm as our base policy π_{base} in conjunction with the `SAGE` meta-algorithm. It will be clear from the sequel that any other `Experts` policy may also be used, such as `Squint` (Koolen and Van Erven, 2015) and `AdaHedge` (Erven et al., 2011), leading to more refined regret bounds.

1. The Base Policy: We start with the standard meta-experts framework as discussed in the Introduction. Define a collection of $\binom{N}{k}$ experts, each corresponding to a distinct k-set 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 the symbol y_t at time t is given by:

$$\begin{aligned} v(y_t, \boldsymbol{p}_t) &= \mathbb{E}\left[\sum_{S:y_t\in S} 1 \times \mathbb{1}(S_t = S) + \sum_{S:y_t\notin S} 0 \times \mathbb{1}(S_t = S)\right] \\ &= \mathbb{P}(y_t \in S_t) \overset{(a)}{=} p_t(y_t), \end{aligned} \tag{8}$$

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` algorithm as our base policy for the resulting `Experts` problem. Let the indicator variables $r_\tau(i) = \mathbb{1}(y_\tau = i), \forall i \in [N]$, encode the symbol chosen by the adversary at round τ. Furthermore, let the variable $r_\tau(S) \equiv \sum_{i\in S} r_\tau(i)$ 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 notation, 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 chooses the expert S at round t with the following probability:

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

where $w_\tau(S) \equiv \exp(\eta R_\tau(S)), \eta > 0$ is the learning rate.

2. Efficient Computation of the Inclusion Probabilities: The marginal inclusion probabilities for each of the N elements can be obtained by marginalizing the probability in Eqn. (9):

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

where $w_{t-1}(i) = \exp(\eta R_{t-1}(i))$. Clearly, for any $S \subseteq [N] \setminus \{i\}$, we have $w_{t-1}(i)w_{t-1}(S) = w_{t-1}(S \cup \{i\})$. We can verify that

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

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, Algorithm 2 may be used to efficiently sample the predicted k-set. However, naïvely computing the probability $p_t(i)$'s directly from Eqn. (10) requires evaluating sums of $\binom{N-1}{k-1}$ products - a computationally intractable task. 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 follows:

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

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 sub-vector 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 deduced in Section A.2 of the supplementary material.

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

Algorithm 3 k**-sets** via `SAGE` with π_{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**

4.2 Regret Bounds

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

$$\text{Regret}_T \leq \sqrt{2 l_T^* \ln n} + \ln n, \tag{13}$$

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

$$\text{Regret}_T(\boldsymbol{y}) \leq \sqrt{2k l_T^*(\boldsymbol{y}) \ln(Ne/k)} + k \ln(Ne/k), \tag{14}$$

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

4.3 k-`sets` with FTRL

It is possible to design efficient online policies for the k-`sets` problem with a base policy other than `Hedge`. In particular, we now show how the standard *Follow-the-Regularlized-Leader* framework can be augmented with the systematic sampling schemes to design an online prediction policy for a generalized version of the k-experts problem with the sum-reward function. Since this policy is not essential to the rest of the paper, due to space limitations, we move this discussion to Section A.3 of the supplementary section. A drawback of the FTRL approach is that, unlike Hedge, this policy does not yield a small-loss regret bound.

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 pairwise-reward function and binary rewards (see Table 1)[1]. Unlike the k-sets problem, where the adversary chooses a single item at each round (so that only one component of the reward vector $\boldsymbol{r}_t$ is one, the rest are zero), here, the adversary secretly chooses a *pair* of items at each round (so that exactly two components of the reward vector $\boldsymbol{r}_t$ are one, the rest are zero). If *both* the items chosen by the adversary are included in the predicted k-set, the learner receives 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 $e_1, e_2, \ldots, e_T$ (arranged in some arbitrary order). 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.

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 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), we get 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

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

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

6 Lower Bounds

In this section, we consider regret lower bounds for different variants of the k-experts problem as given in Table 1. 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:

[1] The general case with arbitrary rewards can be handled using a similar FTRL approach as in Section 4.3.

Theorem 3 (Regret Lower bound for `Sum-reward`). *For any online policy with* $\frac{N}{k} \geq 2$*, we have*

$$\mathcal{R}_T^{\texttt{Sum-reward}} \geq \sqrt{\frac{kT}{2\pi}} - \Theta\left(\frac{1}{\sqrt{T}}\right), \; T \geq 1.$$

Note that with the above rewards structure, the `Sum-reward`, the `Max-reward`, and the `Pairwise-reward` (where the same file is requested twice in each round) variants of the k-`experts` problem become identical. Hence, Theorem 3 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, it can be immediately seen 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 gets rid of the above logarithmic gap.

Theorem 4 (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*

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

A distinguishing feature of the above regret lower bound, compared to the standard lower bounds (Cesa-Bianchi and Lugosi, 2006) 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. This nonlinearity complicates the computation of the optimal expected cumulative reward in hindsight. To overcome this difficulty, we first partition the pool of N experts into k disjoint subsets. We select the cumulative best expert in hindsight from each subset in order to lower bound the optimal offline reward. Please see Section A.4 in the supplementary material for detailed proof.

7 Numerical Experiments

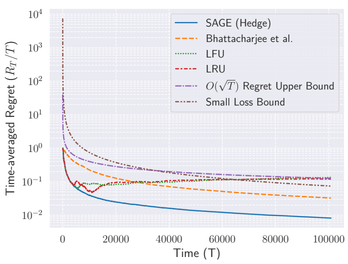

Figure 1: Comparison among different prediction policies with $k/N = 0.1, N \sim 2400$ for the MovieLens Dataset.

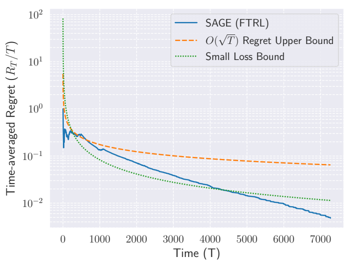

Figure 2: Performance of the `SAGE` policy for pairwise predictions with $k/N = 0.02$ for the Reality Mining Dataset.

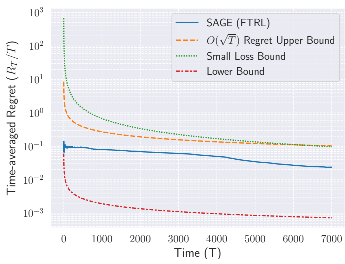

Figure 3: Performance of `SAGE` for the `Max-Reward` function, with $k/N = 0.01$ for the MovieLens Dataset.

k-`sets`: In our first experiment, we consider the following stylized version of the k-`sets` problem. Assume that there is a corpus of N movies. The user may choose to watch 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. Here, we assume that a user rates a movie immediately after watching it. The plot in Figure 1 compares the regrets of the proposed `SAGE` policy (with $\pi_{\text{base}} =$ `Hedge`), the

FTPL policy proposed in 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` decisively outperforms all other policies under consideration.

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 approximate prediction policy 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 $\pi_{\text{base}} = \texttt{FTRL}$. The normalized regret achieved by the policy is plotted in Figure 2. To compute the optimal static offline reward, we used a brute-force search. From the plots, we see that the time-averaged regret for this policy approaches zero for long-enough time-horizon T. This result is quite remarkable in view of the hardness result given in Proposition 1.

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 now 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 predict a closely *related* movie than a completely different one. In Figure 3, we plot the normalized regret of the `SAGE` policy with $\pi_{\text{base}} = \texttt{FTRL}$, along with the lower bound given in Theorem 4. 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, not necessarily for the `Max-reward` function. As of now, we are not aware of any efficient policy for the `Max-reward` version of the k-`experts` problem. Please refer to Section A.5 of the supplement for additional experimental results.

8 Conclusion and Future Research

In this paper, we formulated the k-`experts` problem and designed efficient online policies for some of its variants using the `SAGE` framework. We also derived a tight regret lower bound for the `Max-reward` variant of the k-`experts` problem and characterized the set of all stable mistake bounds for the k-`sets` problem achievable by online policies. However, many interesting questions are still left open. In particular, designing an efficient online policy for the `Max-reward` variant of the problem achieving a regret close to the lower bound given in Theorem 4 will be of interest.

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.

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.

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.

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.

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

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.

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.

A Supplementary Material

A.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

$$\phi_T(\boldsymbol{y}) = \phi(\boldsymbol{y}). \tag{15}$$

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 $\boldsymbol{y} \in [N]^T$:

$$\mathbb{E}\bigg(\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}(y_t \notin S_t)\bigg) = \sum_{t=1}^{T}\big(\phi_t(\boldsymbol{y}^t) - \phi_{t-1}(\boldsymbol{y}^{t-1})\big) + \phi_0, \tag{16}$$

where the above equation follows from telescoping the summation and using the boundary condition (15). 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

$$\mathbb{P}(y_t \notin S_t|\boldsymbol{y}^{t-1}) = 1 - \sum_{i=1}^{N} p_{ti}(\boldsymbol{y}^{t-1})\mathbb{1}(y_t = i). \tag{17}$$

Hence, combining equations (16) and (17), 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 $\boldsymbol{y} \in [N]^T$:

$$\sum_{t=1}^{T}\bigg(-\sum_{i=1}^{N}\frac{p_{ti}(\boldsymbol{y}^{t-1})\mathbb{1}(y_t = i)}{T} + \phi_{t-1}(\boldsymbol{y}^{t-1}) - \phi_t(\boldsymbol{y}^t) + \frac{1}{T}(1-\phi_0)\bigg) = 0. \tag{18}$$

We now consider the following candidate sequence of potential functions:

$$\phi_t(\boldsymbol{y}_t) \equiv \mathbb{E}\phi(\boldsymbol{y}_t, \epsilon_{t+1}^T), \quad \forall t, \tag{19}$$

where the expectation is taken over a random sequence $\boldsymbol{\epsilon}_{t+1}^T$ such that each component $\epsilon_j, t{+}1 \leq j \leq N$ is distributed i.i.d. uniformly over the set $[N]$. It is easy to see that, the boundary condition (15) 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 (18) is satisfied. For, this, we set each of the terms of the equation (18) 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 $\boldsymbol{y}^{t-1} \in [N]^{t-1}$:

$$p_{ti}(\boldsymbol{y}^{t-1}) = T\Big(\phi_{t-1}(\boldsymbol{y}^{t-1}) - \phi_t(\boldsymbol{y}^{t-1}i)\Big) + \frac{k}{N}, \quad \forall i \in [N]. \tag{20}$$

From the definition (19), 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 (20) 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. (20) 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 $\boldsymbol{y}^{t-1}$:

$$-\frac{k}{N} \leq T\bigg(\frac{1}{N}\sum_{i=1}^{N}\phi_t(\boldsymbol{y}^{t-1}i) - \phi_t(\boldsymbol{y}^{t-1}i)\bigg) \leq 1 - \frac{k}{N}. \tag{21}$$

It immediately follows that the stability conditions, given by equations (6) and (7), are sufficient to ensure the bound in Eqn. (21).

Phase-II: Sampling the Predicted set We use the conditional marginal inclusion probabilities $\boldsymbol{p}_t$, derived in Eqn. (20), 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.

A.2 Iterative evaluation of the marginal inclusion probabilities

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

$$g_t(X) = \prod_{i \in [N]} (X - w_t(i)) = \sum_{j=0}^{N} a_{tj} X^j, \tag{22}$$

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,

$$\begin{aligned} \sum_{1 \leq i_1 < i_2 < \cdots < i_k \leq N} \prod_{j=1}^{k} w_t(i_j) &= (-1)^k a_{t,N-k} \\ \iff \sum_{S' \subset [N]: |S'| = k} w_t(S') &= (-1)^k a_{t,N-k}. \end{aligned} \tag{23}$$

Now define

$$g_{ti}(X) = \frac{g_t(X)}{X - w_t(i)} = \sum_{j=0}^{N-1} b_{tj}^{(i)} X^j, \tag{24}$$

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,

$$\sum_{S \subset [N] \setminus \{i\}: |S| = k-1} w_t(S) = (-1)^{k-1} b_{t,N-k}^{(i)}. \tag{25}$$

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

$$p_t(i) = -\frac{w_{t-1}(i) b_{t-1,N-k}^{(i)}}{a_{t-1,N-k}}, \ \forall i \in [N]. \tag{26}$$

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,

$$\begin{aligned} g_{ti}(X) &= -w_t^{-1}(i) g_t(X) \sum_{l \geq 0} X^l w_t^{-l}(i) \\ &= -w_t^{-1}(i) \sum_{j=0}^{N} \sum_{l=0}^{\infty} a_{tj} w_t^{-l}(i) X^{j+l}. \end{aligned} \tag{27}$$

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

$$b_{tj}^{(i)} = -\sum_{l=0}^{j} a_{tl} w_t^{-(j-l+1)}(i) \tag{28}$$

Consequently, we can further express the probability selection rule from Eq. (26) as

$$p_t(i) = \frac{\sum_{j=0}^{N-k} a_{t-1,j} w_{t-1}^{-(N-k-j)}(i)}{a_{t-1,N-k}}. \tag{29}$$

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,

$$
\begin{aligned}
g_t(X) &= \prod_{i=1}^{N}(X - w_t(i)) = g_{t-1}(X) \cdot \frac{X - w_t(f_t)}{X - w_{t-1}(f_t)} \\
&= g_{t-1,f_t}(X)(X - w_t(f_t)) = \sum_{j=0}^{N-1} b^{(f_t)}_{t-1,j} X^j (X - w_t(f_t)).
\end{aligned} \tag{30}
$$

Therefore, using the above and the update rule of $b^{(i)}_{tj}$ from Eq. (28), we obtain,

$$
\begin{aligned}
a_{tj} &= b^{(f_t)}_{t-1,j-1} - w_t(f_t) b^{(f_t)}_{t-1,j} \\
&= w_t(f_t) \sum_{k=0}^{j} a_{t-1,k} w_{t-1}^{-(j-k+1)}(f_t) - \sum_{k=0}^{j-1} a_{t-1,k} w_{t-1}^{-(j-k)}(f_t) \\
&= (e^\eta - 1) \sum_{k=0}^{j} a_{t-1,k} w_{t-1}^{-(j-k)}(f_t) + a_{t-1,j},
\end{aligned} \tag{31}
$$

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. (31) can be used to obtain a further simplified recurrence to the update of the coefficients $a_{t,i}$ as below:

$$
\begin{aligned}
a_{t,j} &= (e^\eta - 1) w_{t-1}^{-1}(f_t) \sum_{k=0}^{j-1} a_{t-1,k} w_{t-1}^{-(j-1-k)}(f_t) + e^\eta a_{t-1,j}, \\
&= w_{t-1}^{-1}(f_t)\big(a_{t,j-1} - a_{t-1,j-1}\big) + e^\eta a_{t-1,j}, \ \forall 1 \leq j \leq N, && (32) \\
a_{t,0} &= e^\eta a_{t-1,0}. && (33)
\end{aligned}
$$

Using the update equations of $\{a_{tj}\}_{j=1}^N$ and $\{p_t(j)\}$ from Eqs. (32), (33) and (29) respectively, we have the following iterative numerical procedure for computing the marginal inclusion probabilities:

Algorithm 5 Iterative Computation of the Marginal Inclusion Probabilities

Input: Learning rate $\eta > 0$,

Initialize: $\boldsymbol{R}_0 = \mathbf{0}, a_{0,j} = (-1)^{N-j}\binom{N}{j}, \ \forall 0 \leq j \leq N.$

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 $y_{t,i} = \mathbb{1}\{i \in S_t\}$.

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

$$
\begin{aligned}
a_{t,0} &\leftarrow e^\eta a_{t-1,0} \\
a_{t,j} &\leftarrow w_{t-1}^{-1}(f_t)\big(a_{t,j-1} - a_{t-1,j-1}\big) + e^\eta a_{t-1,j}, \ 1 \leq j \leq N.
\end{aligned}
$$

6: **end for**

A.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.,*

$$\mathcal{R}_T := \max_{\boldsymbol{p}^* \in \Delta(\mathcal{C}_k^N)} \sum_{t=1}^{T} \psi(\boldsymbol{r}_t \cdot \boldsymbol{p}^*) - \sum_{t=1}^{T} \psi(\boldsymbol{r}_t \cdot \boldsymbol{p}_t), \tag{34}$$

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. (34) 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:

$$\boldsymbol{p}_t = \arg\max_{\boldsymbol{p} \in \Delta_k^N} \left[\Big(\sum_{s=1}^{t-1} \nabla_s\Big)^T \boldsymbol{p} - \frac{1}{\eta} \sum_{i=1}^{N} p_i \ln p_i, \right] \tag{35}$$

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 (35) may be quickly determined in $\tilde{O}(N)$ time as shown in Algorithm 6 below.

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**

See Section A.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 (35) 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 5 (Regret Bound)**.** *With the learning rate $\eta > 0$, the **FTRL** policy for the generalized **k-sets** problem with the entropic regularizer ensures that*

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

where $||\nabla_{t,i}^2||_{k,\infty}$ denotes the sum of the k largest components of the vector ∇_t^2, 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.*,

$$\boldsymbol{x}_{t+1} = \arg\max_{\boldsymbol{x}\in\Omega} \left[(\sum_{s=1}^{t} \nabla_s^T)\boldsymbol{x} - \frac{1}{\eta} g(x), \right] \tag{36}$$

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:

$$\text{Regret}_T^{\text{FTRL}} \leq 2\eta \sum_{t=1}^{T} ||\nabla_t||_{*,t}^2 + \frac{g(\boldsymbol{u}) - g(\boldsymbol{x}_1)}{\eta}, \tag{37}$$

where the quantity $||\nabla_t||_{*,t}^2$ 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

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

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

To bound the second term in (37), 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 (37) can be simplified as follows:

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

□

A.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 (35) 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 (35) 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,

$$\sum_{i=1}^{N} p_i = k \tag{38}$$
$$p_i \leq 1, \quad \forall i \tag{39}$$
$$p_i \geq 0, \quad \forall i. \tag{40}$$

By associating the real variable λ with the constraint (38) and the non-negative dual variable ν_i with the i^{th} constraint in (39), we construct the following Lagrangian function:

$$L(\boldsymbol{p}, \lambda, \boldsymbol{\nu}) = \sum_i \big(p_i R_{t-1,i} - \frac{1}{\eta} p_i \ln p_i - \lambda p_i - \nu_i p_i\big) \tag{41}$$

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 (39), 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 (38):

$$\sum_{i=1}^{N} \min(1, K \exp(\eta R_{t-1,i})) = k. \tag{42}$$

For any $k < N$, we now argue that the equation (42) has a unique solution for $K > 0$. The LHS of the equation (42) is a continuous, non-decreasing function of K and takes value in the interval $[0, N]$. Hence, by the intermediate value theorem, the equation (42) 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 (42), 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 (42) can be written as:

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

i.e.,

$$K = \frac{k - i^*}{\sum_{j=i^*+1}^{N} \exp(\eta R_{t-1,(j)})}. \tag{43}$$

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

$$(k - i^*) \exp(\eta R_{t-1,(i^*)}) \geq \sum_{j=i^*+1}^{N} \exp(\eta R_{t-1,(j)}). \tag{44}$$

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 (44) is satisfied. Once the optimal i^* is found, the optimal value of the constant K may be obtained from equation (43). 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.

A.4 Proof of Theorem 4

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 [2]. 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 4 finally follows from a tight non-asymptotic lower bound to this expectation, which we believe, has not appeared in this form before.

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

$$\max_{\{\boldsymbol{r}_t\}_{t=1}^T} \mathcal{R}_T \geq \mathbb{E}_r\big(\mathcal{R}_T\big), \tag{45}$$

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. $\sim$ 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:

$$\max_{\{\boldsymbol{r}_t\}_{t=1}^T} \mathcal{R}_T \geq \texttt{OPT} - \sum_{t=1}^T \mathbb{E}_r\big(\max_{i\in S_t} r_{t,i}\big), \tag{46}$$

where we denote

$$\texttt{OPT} = \mathbb{E}_r\big(\max_{S\subset[N]:|S|=k} \sum_{t=1}^T \max_{i\in S} r_{t,i}\big). \tag{47}$$

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

$$\mathbb{E}_r\big(\max_{i\in S_t} r_{t,i}\big) = \mathbb{P}\big(\max_{i\in S_t} r_{t,i} = 1\big) = 1 - (1-p)^k. \tag{48}$$

It now remains to establish a lower bound to the quantity `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:

$$\texttt{OPT} \geq \sum_{t=1}^T \mathbb{E}\big(\max_{i\in S} r_{t,i}\big). \tag{49}$$

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 (49) 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

$$B_l = \{(l-1)b + 1, \cdots, lb\}, \quad 1 \leq l \leq k. \tag{50}$$

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

[2]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.

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

$$q \equiv \mathbb{E}(r_{t,i_l}) = \frac{1}{T}\mathbb{E}(X_{T,i_l}) = \frac{1}{T}\mathbb{E}(\max_{j \in B_l} X_{T,j}). \tag{51}$$

Hence, using the lower bound (49), we have

$$\texttt{OPT} \geq \sum_{t=1}^{T} \left(1 - (1-q)^k\right) = T(1 - (1-q)^k). \tag{52}$$

Hence, combining Eqns. (46), (48) with the lower bound in Eqn. (52), we have the following regret lower bound in terms of the yet undetermined parameter q:

$$\max_{\{\boldsymbol{r}_t\}_{t=1}^T} \mathcal{R}_T \geq T\left((1-p)^k - (1-q)^k\right). \tag{53}$$

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

$$\max_{\{\boldsymbol{r}_t\}_{t=1}^T} \mathcal{R}_T \geq kT(q-p)(1-q)^{k-1}. \tag{54}$$

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:

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

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:

$$q \leq \frac{3p}{2}. \tag{55}$$

2. Lower bounding q: We have

$$\begin{aligned} q - p &= \frac{1}{T}\mathbb{E}\left(\max_{j \in B_1}(X_{T,j} - Tp)\right) \\ &= \frac{1}{T}\mathbb{E}\left(\max_{j \in B_1}(X_{T,j} - Tp)\mathbb{1}\left\{\max_{j \in B_1} X_{T,j} < Tp\right\}\right) \\ &+ \frac{1}{T}\mathbb{E}\left(\max_{j \in B_1}(X_{T,j} - Tp)\mathbb{1}\left\{\max_{j \in B_1} X_{T,j} \geq Tp\right\}\right) \\ &\stackrel{\text{(def.)}}{=} \frac{I_1 + I_2}{T}. \end{aligned} \tag{56}$$

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:

$$\begin{aligned}
I_1 &\equiv \mathbb{E}\bigg(\max_{j\in B_1}(X_{T,j}-Tp)\mathbb{1}\left\{\max_{j\in B_1}X_{T,j}<Tp\right\}\bigg) \\
&\overset{(a)}{\geq} \max_{j\in B_1}\mathbb{E}\bigg((X_{T,j}-Tp)\mathbb{1}\left\{X_{T,j}<Tp\right\}\prod_{i\in B_1,i\neq j}\mathbb{1}\left\{X_{T,i}<Tp\right\}\bigg) \\
&\overset{(b)}{=} \mathbb{E}\bigg((X_{T,1}-Tp)\mathbb{1}\left\{X_{T,1}<Tp\right\}\bigg)\bigg(\mathbb{P}(X_{T,1}<Tp)\bigg)^{b-1} \\
&\overset{(c)}{\geq} -\mathbb{E}\big|X_{T,1}-Tp\big|\bigg(\mathbb{P}(X_{T,1}<Tp)\bigg)^{b-1} \\
&\overset{(d)}{\geq} -\sqrt{Tp(1-p)}\bigg(\mathbb{P}(X_{T,1}<Tp)\bigg)^{b-1} \\
&\overset{(e)}{\geq} -\big(\frac{3}{4}\big)^{b-1}\sqrt{Tp(1-p)}. \qquad (57)
\end{aligned}$$

in the above,

1. inequality (a) follows from Jensen's inequality and the trivial fact that $\mathbb{1}\left\{\max_{j\in B_1}X_{T,j}<Tp\right\} = \mathbb{1}\left\{X_{T,j}<Tp\right\}\prod_{i\in B_1,i\neq j}\mathbb{1}\left\{X_{T,i}<Tp\right\}$
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 $(X_{T,1}-Tp)\mathbb{1}\left\{X_{T,1}<Tp\right\} \geq -\big|X_{T,1}-Tp\big|$,
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 $s\geq 0$:

$$\begin{aligned}
I_2 &\geq s\mathbb{P}\big(\max_{j\in B_1}X_{T,j} > s+Tp\big) \\
&\overset{(a)}{=} s\left(1-\big(\mathbb{P}\big(X_{T,1}\leq s+Tp\big)\big)^b\right) \\
&\overset{(b)}{\geq} s\bigg(1-\big(\Phi(\frac{s}{\sqrt{Tp(1-p)}})\big)^b\bigg). \qquad (58)
\end{aligned}$$

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:

$$\begin{aligned}
\Phi(u) &= 1-\frac{1}{\sqrt{2\pi}}\int_u^\infty e^{-x^2/2}dx \\
&\leq 1-\frac{1}{\sqrt{2\pi}}\int_u^{2u} e^{-x^2/2}dx \\
&\leq 1-\frac{ue^{-2u^2}}{\sqrt{2\pi}}. \qquad (59)
\end{aligned}$$

By making a change of variable $u\leftarrow \frac{s}{\sqrt{Tp(1-p)}}$ in Eqn. (58), the quantity I_2 can be lower bounded as:

$$I_2 \geq \sqrt{Tp(1-p)}\bigg[u\bigg(1-\big(1-\frac{ue^{-2u^2}}{\sqrt{2\pi}}\big)^b\bigg)\bigg]. \qquad (60)$$

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

$$I_2 \geq c_1 \sqrt{Tp(1-p)\ln b}, \tag{61}$$

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 (57) and (61), we obtain the following lower bound for q from Eqn. (56) valid for $b \equiv \frac{N}{k} \geq 7$:

$$q - p \geq \frac{c_2}{T}\sqrt{Tp(1-p)\ln\frac{N}{k}}, \tag{62}$$

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 (55) and (62) into the regret lower bound (54), for $T \geq 16k\ln(\frac{N}{k})$ and $\frac{N}{k} \geq 7$, we obtain:

$$\max_{\{\boldsymbol{r}_t\}_{t=1}^T} \mathcal{R}_T \geq c_2 k \sqrt{\frac{T}{2k}(1 - \frac{1}{2k})\ln\frac{N}{k}}\left(1 - \frac{3}{4k}\right)^{k-1} \geq c_3\sqrt{kT\ln\frac{N}{k}}, \tag{63}$$

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

A.5 Additional Experimental Results

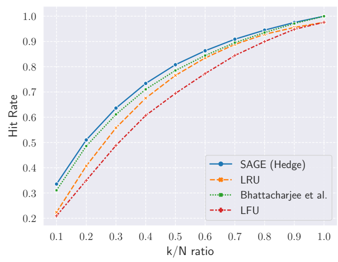

Figure 4: Comparison among different prediction policies in terms of hit rates (fraction of correct predictions) for different values of k/N, $N \sim 2400$ for the MovieLens dataset.

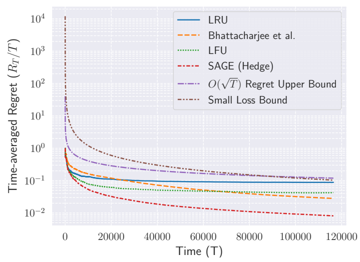

Figure 5: Comparison among different prediction policies in terms of normalized regret $\frac{R_T}{T}$ with $k/N = 0.1$, $N \sim 2500$ for the Wiki-CDN dataset.

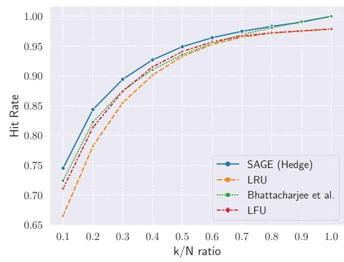

Figure 6: Comparison among different prediction policies in terms of hit rates (fraction of correct predictions) for different values of k/N, $N \sim 2500$ for the Wiki-CDN dataset.

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