A Test Score Based Approach to Stochastic Submodular Optimization

Shreyas Sekar
Harvard Business School, Boston, MA, ssekar@hbs.edu

Milan Vojnovic
Department of Statistics, London School of Economics (LSE), London, UK, m.vojnovic@lse.ac.uk

Se-Young Yun
Department of Industrial and System Engineering, KAIST, South Korea, yunseyoung@kaist.ac.kr

We study the canonical problem of maximizing a stochastic submodular function subject to a cardinality constraint, where the goal is to select a subset from a ground set of items with uncertain individual performances to maximize their expected group value. Although near-optimal algorithms have been proposed for this problem, practical concerns regarding scalability, compatibility with distributed implementation, and expensive oracle queries persist in large-scale applications. Motivated by online platforms that rely on individual item scores for content recommendation and team selection, we study a special class of algorithms that select items based solely on individual performance measures known as test scores. The central contribution of this work is a novel and systematic framework for designing test score based algorithms for a broad class of naturally occurring utility functions. We introduce a new scoring mechanism that we refer to as *replication test scores* and prove that as long as the objective function satisfies a diminishing returns condition, one can leverage these scores to compute solutions that are within a constant factor of the optimum. We then extend these scoring mechanisms to the more general stochastic submodular welfare maximization problem, where the goal is to partition items into groups to maximize the sum of the expected group values. For this more difficult problem, we show that replication test scores can be used to develop an algorithm that approximates the optimum solution up to a logarithmic factor. The techniques presented in this work bridge the gap between the rigorous theoretical work on submodular optimization and simple, scalable heuristics that are useful in certain domains. In particular, our results establish that in many applications involving the selection and assignment of items, one can design algorithms that are intuitive and practically relevant with only a small loss in performance compared to the state-of-the-art approaches.

Key words: stochastic combinatorial optimization, submodular functions, welfare maximization, test scores

1. Introduction

A common framework for combinatorial optimization that captures problems arising in wide-ranging applications is that of *selecting a finite set of items* from a larger candidate pool and *assigning these items to one or more groups*. Such problems form the core basis for online content recommendation systems (Agarwal et al. 2008, Besbes et al. 2015), e-commerce (Li 2011), digital advertising (Mehta et al. 2013) and project portfolio selection (Chien 2002) as well as *team* *selection* problems arising in online gaming (Herbrich et al. 2006) and traditional hiring. A crucial feature of these environments is the intrinsic uncertainty associated with the underlying items and consequently, sets of items. Given this uncertainty, the decision maker's objective in these domains is to maximize the expected group value associated with the set of items and their assignment.

As a concrete application, consider an online gaming platform where the items correspond to players; the platform may seek to assign a subset of players to teams in order to ensure competitive matches or to maximize the winning probability for a specific team. Other scenarios relating to team selection—e.g., a company hiring a set of candidates or a school identifying top students for a tournament—can also be modeled in an analogous fashion. Alternatively, these optimization problems arise in online communities such as Stack Overflow or Reddit. Here, the items represent topics or questions and the platform wishes to present a collection of relevant topics to an incoming user with the goal of maximizing that user's engagement measured via clicks or answers. Finally, in project selection, items may refer to potential R&D efforts and the firm may be interested in green-lighting a limited portfolio of projects. Naturally, all of these constitute stochastic environments due to the underlying uncertainty, e.g., (*i*) the performance of any individual player is not deterministic in the case of a gaming platform, (*ii*) there is considerable uncertainty regarding a user's propensity to click or respond to a topic on knowledge platforms, and (*iii*) R&D projects are intrinsically risky and their precise output cannot be predicted in advance.

There are several fundamental challenges in the above applications that necessitate innovative algorithmic approaches. First, the value derived from a set of items may not be linear in that of the individual items and may in fact, model a more subtle relationship. For example, agents or topics may complement or supplement each other; the efficiency of a team may grow with team size but exhibit diminishing returns as more members are added due to coordination inefficiencies. Second, the intrinsic uncertainty regarding the value of individual items may affect the group value in intricate ways due to the non-linearity of the objective. As we depict later, there are situations where a set of 'high-risk high-reward' items may outperform a collection of stable-value items even when the latter type provides higher value in expectation. Finally, we also face issues relating to computational complexity since the number of items and groups can be very large in online platform scenarios and the underlying combinatorial optimization problems are usually NP-hard.

Despite the above challenges, a litany of sophisticated algorithmic solutions have been developed for the problems mentioned previously. Given the intricacies of the setting, these algorithms tend to be somewhat complex and questions remain on whether these methods are suitable for the scenarios outlined earlier owing to issues regarding scalability, interpretability, and the difficulties of function evaluation (Agrawal et al. 2018, Chien 2002). On the other hand, it is common practice in many domains to select or assign items by employing algorithms that base their decisions on *individual item scores*—these represent unique statistics associated with each item that serve as a proxy for the item's quality or the relevance to the task at hand (Henriksen and Traynor 1999, Koç and Morton 2014). At a high level, these algorithms only use the scores computed for individual items—each item's score is independent of other items—and as such, avoid the practical issues that plague traditional algorithmic paradigms.

To expand on this thesis, consider a dynamic online portal such as Stack Overflow that hosts over eighteen million questions and wishes to recommend the most relevant subset to each incoming user. The platform may find it impractical to recompute the optimal recommended set of questions every time a new batch of questions is posted and thus, many traditional optimization methods are not scalable (Besbes et al. 2015). At the same time, content recommendation services typically maintain relevance scores for each item and user pair that do not vary as new questions are posted and are utilized in practice to generate recommendation sets (Agarwal et al. 2008, Manning et al. 2010). In a similar vein, online gaming platforms estimate skill ratings (scores) for individual players based only on their past performance, which are in turn used as inputs for matchmaking (Herbrich et al. 2006, Yuan et al. 2007). When it comes to team formation, these score based approaches may be preferable to standard algorithms that require *oracle access* to the performance of every possible team. Indeed, evaluating the expected value of every subset of players even before the teams are formed seems prohibitively expensive. Finally, in the case of project selection, firms may prefer to invest in R&D undertakings based on their individual risk-reward profile, which is typically estimated via a scoring function (Chien 2002, Henriksen and Traynor 1999).

Clearly, algorithms for selecting or assigning items based solely on individual item scores are appealing in many domains because of their conceptual and computational simplicity. However, a natural concern is that restricting the algorithmic landscape to these simple score based approaches may result in suboptimal solutions because they may be unable to account for complicated dependencies between individual performance and the group output. Motivated by this tension, we study the following fundamental question:

> *Can algorithms that assign items to groups based on individual item scores achieve near-optimal group performance and if so, under what conditions?*

We briefly touch upon our framework for stochastic combinatorial optimization. Let $N = \{1, 2, \dots, n\}$ be a ground set of items and let 2^N denote all possible subsets of N. Given a feasible set $\mathcal{F} \subseteq 2^N$ of items, a value function $f : 2^N \times \mathbf{R}^n \to \mathbf{R}_+$, and a distribution P of a random n-dimensional vector $\mathbf{X} = (X_1, X_2, \dots, X_n)$, our goal is to select a set $S^* \in \mathcal{F}$ that is a solution to

$$\max_{S \in \mathcal{F}} u(S) := \mathbf{E}_{\mathbf{X} \sim P}[f(S, \mathbf{X})]. \tag{1}$$

In later sections, we generalize this formulation to consider problems where the goal is to select multiple subsets of N and assign them to separate groups. The optimization problem (1) is further refined as follows (see Section 2 for formal definitions):

(a) We focus primarily on the canonical problem of *maximizing a stochastic monotone submodular function subject to a cardinality constraint.* This is a special case of the optimization problem in (1) where $\mathcal{F}$ is defined by the cardinality constraint $|S| = k$ for a given parameter k, and value function f is such that the set function $u: 2^N \rightarrow \mathbf{R}_+$ is submodular.

(b) We restrict our attention to value functions f where the output of $f(S, \mathbf{X})$ depends only on the elements of $\mathbf{X}$ that correspond to S, i.e., $\{X_i\}_{i \in S}$. Further, X_i denotes the random performance of item $i \in N$ and is distributed independently of all other X_j for $j \neq i$. Therefore, $P(x_1, x_2, \ldots, x_n) = P_1(x_1)P_2(x_2)\cdots P_n(x_n)$ where $P_i(x_i) = \mathbf{Pr}[X_i \leq x_i]$, for $i = 1, 2, \ldots, n$.

The framework outlined above captures a broad class of optimization problems arising in diverse domains. For example, submodular functions were featured in a variety of applications such as facility location (Ahmed and Atamtürk 2011), viral influence maximization (Kempe et al. 2015), job scheduling (Cohen et al. 2019), content recommendation and team formation (Bhowmik et al. 2014). In particular, submodularity allows us to model positive synergies among items and capture the natural notion of *diminishing returns to scale* that is prevalent in so many situations—i.e., the marginal value derived by adding an item to a set cannot be greater than that obtained by adding it to any of its subsets. Moreover, in content recommendation as well as team selection, it is natural to expect that the performance of a group of elements S would simply be a function (albeit a non-linear one) of the individual performances of the members in S, i.e. $\{X_i\}_{i \in S}$. This is represented by our assumptions on the value function f.

The problem of maximizing a submodular function subject to a cardinality constraint is known to be NP-hard and consequently, there is a rich literature on approximation algorithms for both the deterministic (Krause and Golovin 2014) and stochastic variants (Asadpour and Nazerzadeh 2016). In a seminal paper, Nemhauser et al. (1978) established that a natural greedy algorithm (sequentially selecting items that yield largest marginal value) guarantees a $1 - 1/e$ approximation of the optimum value, which is tight under the value oracle model (Feige 1998). Despite the popularity of greedy and other approaches, it is worth noting for our purposes that almost all of the algorithms in this literature are *not robust to changes in the input.* That is, as the ground set N grows, it is necessary to re-run the entire greedy algorithm to generate an approximately optimal subset. As mentioned earlier, these methods also extensively utilize *value oracle queries*—access to the objective function is through a black-box returning $u(S)$ for any given set S.

Test Score Algorithms We now formalize the notion of individual item scores, which we refer to as *test scores*. Informally, a test score is an item-specific parameter that quantifies the suitability of the item for the desired objective (i.e., f). To ensure scalability, it is crucial that an item's score depends only on the marginal distribution the item's individual performance and the problem specification. Formally, the test score $a_i \in [0, \infty)$ of an item $i \in N$ is defined as:

$$a_i = h(f, \mathcal{F}, P_i), \tag{2}$$

where h is a mapping from the item's marginal distribution P_i, the objective value function f and constraint set $\mathcal{F}$ to a single number. Naturally, there are innumerable ways to devise a meaningful test score mapping h. Obvious examples include: (a) *mean test scores* where $a_i = \mathbf{E}[X_i]$, and (b) *quantile test scores*, where a_i is the θ-quantile of distribution P_i, i.e. $a_i = \inf\{x \in \mathbf{R} \mid P_i(x) \geq \theta\}$, for a fixed value $\theta \in [0, 1]$. However, we prove later that algorithms that base their decisions on these natural candidates do not always yield good solutions.

The design question studied in this paper is to identify a suitable test score mapping h such that algorithms that leverage these scores can obtain near-optimal guarantees for the problem defined in (1). Formally, a test score algorithm is a procedure that computes the test scores for each item in N according to some mapping h and uses only these scores to determine a feasible solution S for (1), e.g., by selecting the k items with the highest scores. Test score algorithms were first introduced by Kleinberg and Raghu (2018), who developed algorithms for a team formation problem for a specific function f. In this work, we propose a novel test score mechanism and utilize it to retrieve improved guarantees for a large class of naturally occuring functions.

Test score algorithms are particularly salient in large-scale applications when compared to a more traditional optimization method such as greedy or LP-based algorithms. First, as the ground set N changes (e.g., items are added or deleted), this does not alter the scores of items still present in the ground set since an item's test score depends only on its own performance distribution. This allows us to eliminate significant computational overhead in dynamic environments such as online platforms (Agarwal et al. 2008, Besbes et al. 2015). Second, test score computations are trivially parallelizable—implemented via distributed computation—since each item's test score can be computed on a separate machine. Designing algorithms that are amenable to distributed implementation (Balkanski et al. 2019) is a major concern nowadays and it is worth noting that standard greedy or linear programming approaches do not fulfill this criterion. Finally, test score algorithms allow us to make fewer and simpler oracle calls (function evaluations) as we highlight later. We now present a stylized formulation of a stochastic submodular optimization problem in an actual application in order to better illustrate the role of test scores.

Example 1. **(Content Recommendation on Stack Overflow or Reddit)** The ground set N comprises of topics created by users on the website. The platform is interested in selecting a set of k topics from the ground set and presenting them to an arriving user in order to maximize satisfaction or engagement. For simplicity, the topics can be broadly classified into two categories—set A consisting of useful but not very exciting topics and set B which encapsulates topics that are polarizing or exciting[1]. Mathematically, we can capture this selection problem using our framework by taking X_i to denote the utility that a user derives from topic $i \in N$ (alternatively X_i could denote the probability of clicking or responding to a topic). For example, $X_i = a$ with probability 1 for $i \in A$ as these topics are stable, whereas $X_i = b/p$ with probability p for each risky topic $i \in B$. The selection problem becomes particularly interesting when $b < a < b/p$. Due to cognitive limitations, one can assume that a user engages with at most $r \leq k$ topics from the assortment. Therefore, the objective function is defined as follows: $f(S, \mathbf{X}) = \sum_{j=1}^{r} X_{(j)}(S)$, where $X_{(j)}(S)$ refers to the j-th largest variable X_i for $i \in S$. In the extreme case, $r = 1$ and each user clicks on at most one topic. We refer to these as the *top-r* and *best-shot* functions respectively in Section 2. ■

The tradeoff between 'high-risk-high-reward' items and more stable items arises in a large class of selection problems in the presence of uncertainty. For example, in online gaming as in other team selection scenarios, a natural contention occurs between high performing players who exhibit a large variance (set B) and more consistent players (set A). In applications involving team formation, it is natural to use the *CES (Constant Elasticity of Substitution)* utility function as the objective, i.e., $f(S, \mathbf{X}) = (\sum_{i \in S} X_i^r)^{1/r}$, where the value of r indicates the degree of substitutability of the task performed by the players (Fu et al. 2016). In this work, we design a natural test score based algorithm that allows us to obtain constant factor approximations for stochastic submodular optimization for all of the above objectives functions.

1.1. Main Contributions

The primary conceptual contribution of this study is the introduction of a framework for analysis of test score based algorithms for stochastic combinatorial optimization problems involving selection and assignment. We believe that this paradigm helps bridge the gap between theory and practice, particularly in large-scale applications where quality or relevance scores are prominently used for optimization. For these cases, the mechanisms developed in this work provide a rigorous framework for computing and utilizing these scores.

Our main technical contribution is the design of a test score mapping which gives us good approximation algorithms for two NP-hard problems, namely: (a) maximizing a stochastic monotone

[1] For instance, Reddit identifies certain posts as controversial based on the ratio of upvotes and downvotes.

submodular function subject to a cardinality constraint, and (b) maximizing a stochastic submodular welfare function, defined as a sum of stochastic monotone submodular functions subject to individual cardinality constraints. The welfare maximization problem is a strict generalization of the former and is of interest in online platforms, where items are commonly assigned to multiple groups, e.g., selection of multiple disjoint teams for an online gaming tournament.

We now highlight our results for the first problem. We identify a special type of test scores that we refer to as *replication test scores* and show that under a sufficient condition on the value function (extended diminishing returns), we achieve a *constant factor approximation for the problem of maximizing a stochastic submodular function subject to a cardinality constraint.* At a high level, replication test scores can be interpreted as a quantity that measures both an item's individual performance as well its marginal contribution to a team of equally skilled items—see Section 3 for a formal treatment. Additionally, we also show the following:

- We provide an intuitive interpretation of the extended diminishing returns condition and prove that it is satisfied by a number of naturally occuring value functions including but not limited to the functions mentioned in our examples such as best-shot, top-r, and CES.
- We show that replication scores enjoy a special role in the family of all feasible test scores: in particular, for any given value function, if there exist any test scores that guarantee a constant factor approximation for the submodular maximization problem, then it is possible to obtain a constant factor approximation using replication test scores. This has an important implication that in order to find good approximation factors, it suffices to consider replication test scores.
- We highlight cases where natural test score measures such as mean and quantile test scores do not yield a constant factor approximation. We provide a tight characterization of their efficiency for the CES function—specifically, mean test scores provide only a $1/k^{1-1/r}$-approximation to the optimum and quantile scores do not guarantee a constant-factor approximation when $r < r^*(k)$ where $r^*(k) = \Theta(\log(k))$. Recall that r denotes the degree of substitutability among items.

Finally, for the more general problem of stochastic submodular welfare maximization subject to cardinality constraints, with the value functions satisfying the extended diminishing returns condition, we establish that replication test scores guarantee a $\Omega(\frac{1}{\log(k)})$-approximation to the optimum value, where k is the maximum cardinality constraint. This approximation is achieved via a more intricate algorithm that greedily assigns items to groups based on their replication test scores.

Our results are established by a novel framework that can be seen as approximating (sketching) set functions using test scores. In general, a sketch of a set function is defined by two simpler functions that lower and upper bound the original set function within given approximation factors.

In our context, we present a novel construction of a sketch that only relies on replication test scores to approximate a submodular function *everywhere*. By leveraging this sketch, we show that selecting the k items with the highest test scores is only a constant factor smaller than the optimal set. These results may be of independent interest.

1.2. Related Work

The problem of maximizing a stochastic submodular function subject to a cardinality constraint by using test scores was first posed by Kleinberg and Raghu (2018) who developed constant factor approximation algorithms for a specific value function, namely the top-r function. They introduced the term 'test scores' in the context of designing algorithms for team hiring to indicate that the relevant score for each candidate can often be measured by means of an actual test. Their work also provides some impossibility results, namely that test score algorithms cannot yield desirable guarantees for certain submodular functions. Our work differs in several respects. First, we show that test scores can guarantee a constant factor approximation for a broad class of stochastic monotone submodular functions, which includes different instances of value functions used in practice. Second, we extend this theoretical framework to the more general problem of stochastic submodular welfare maximization, and obtain novel approximation results by using test scores. Third, we develop a unifying and systematic framework based on approximating set functions by simpler test score based sketches.

As we touched upon earlier, submodular functions are found in a plethora of settings and there is a rich literature on developing approximation algorithms for different variants of the cardinality-constrained and welfare maximization problems (Lehmann et al. 2006, Vondrak 2008). Commonly used algorithmic paradigms for these problems include greedy, local search, and linear programming (with rounding). Due to their reliance on these sophisticated techniques, most if not all of these algorithms are (a) not scalable in dynamic environments as the algorithm has to be fully re-executed every time the ground set changes, and (b) hard to implement in a parallel computing model. More importantly, these policies are inextricably tied to the value oracle model and hence, tend to query the oracle a large number of times; often these queries are aimed at evaluating the function value for arbitrary subsets of the ground set. As we illustrate in Section 2.4, oracle queries can be expensive in certain cases. On the other hand, the test score algorithm proposed in this work makes use of much fewer oracle queries. Within the realm of submodular maximization, there are three distinct strands of literature that seek to tackle each of the three issues mentioned above.

- *Dynamic Environments*: A growing body of work has sought to develop *online algorithms* for submodular and welfare maximization problems in settings where the elements of the ground set arrive sequentially (Feldman and Zenklusen 2018, Korula et al. 2018) In contrast to this work, the decisions made by online algorithms are irrevocable, where test score algorithms are only aimed at reducing the computational burden when the ground set changes.

- *Distributed Implementation*: Following the rise of big data applications and map-reduce models, there has been a renewed focus on developing algorithms for submodular optimization that are suitable for parallel computing. The state-of-the-art (distributed) algorithms for submodular maximization are $O(\log(n))$-adaptive—they run for $O(\log(n))$ sequential rounds with parallel computations in each round (Balkanski et al. 2019, Ene and Nguyen 2019). Since each test score can be computed independently, our results can be interpreted as identifying a well-motivated special class of submodular functions which admit 1-adaptive algorithms.
- *Oracle Queries*: The design of test score algorithms is well-aligned with the body of work on maximizing submodular set functions using a small number of value oracle queries (Badanidiyuru and Vondrák 2014, Ene and Nguyen 2019). In fact, our replication test score based algorithms only query the function value for subsets comprising of similar or identical items.

Although there is a promising literature pertaining to each of these three challenges, our test score based techniques represent the first attempt at addressing all of them. While many of the above papers propose algorithms for deterministic environments, recently, there has been considerable focus on maximizing submodular functions in a stochastic setting (e.g., Hassidim and Singer 2017, Singla et al. 2016, Asadpour and Nazerzadeh 2016, Gotovos et al. 2015, Kempe et al. 2015, Asadpour et al. 2008). However, the methods presented in these works do not address any of the concerns mentioned earlier and to a large extent, focus explicitly on settings where it is feasible to *adaptively probe* items of the ground set to uncover the realization of their random variables.

More generally, a powerful paradigm for solving stochastic optimization problems as defined in (1) is the technique of *Sample Average Approximation* (SAA) (Kleywegt et al. 2002, Shapiro and Nemirovski 2005, Swamy and Shmoys 2012). These methods are typically employed when the following conditions are applicable, see e.g. Kleywegt et al. (2002): (a) the function $u(S)$ cannot be written in a closed form, (b) the value of the function $f(S,\mathbf{x})$ can be evaluated for every given set S and vector $\mathbf{x}$, and (c) the set $\mathcal{F}$ of feasible solutions is large. The fundamental principle underlying this technique is to generate samples $(\mathbf{x}^{(1)},\dots,\mathbf{x}^{(T)})$ independently from the distribution P and use these to compute the set S^* that is the optimal solution to $\arg\max_{S\in\mathcal{F}} \frac{1}{T}\sum_{i=1}^{T} f(S,\mathbf{x}^{(i)})$.

In addition to the same drawbacks regarding scalability mentioned above, there are other situations where it may be advantageous to use a test score algorithm over SAA methods: (a) when the function f is accessed via a value oracle, a large number of queries may be required to optimize the sample-approximate objective, and (b) even if oracle access is not a concern and the underlying function is rather simple (e.g., best-shot function from Example 1), computing the optimal set S^* may be NP-hard (see Appendix H). Finally, by means of numerical simulations in Section 5.1, we highlight well-motivated scenarios where SAA methods may result in a higher error probability compared to test score algorithms under the same number of samples drawn.

The techniques in our work are inspired by the theory on set function sketching (Goemans et al. 2009, Balcan and Harvey 2011, Cohavi and Dobzinski 2017), and their application to optimization problems (Iyer and Bilmes 2013). While the $\Omega(1/\sqrt{n})$ sketch of Goemans et al. (2009) for general submodular functions does apply to our setting, we are able to provide tighter bounds (the logarithmic bound of Lemma 9) for a special class of well-motivated submodular functions that cannot be captured by existing frameworks such as curvature (Sviridenko et al. 2017). Our approach is also similar in spirit to Iyer and Bilmes (2013), where upper and lower bounds in terms of so-called *surrogate functions* were used for submodular optimization; the novelty in the present work stems from our usage of test scores for function approximation, which are conceptually similar to *juntas* (Feldman and Vondrák 2014). We believe that the intuitive and natural interpretation of test score-based algorithms make them an appealing candidate for other problems as well.

1.3. Organization of the Paper

The paper is structured as follows. Section 2 provides a formal definition of the optimization problems studied in this paper and introduces examples of value functions. Section 3 contains our main results for the problem of maximizing a stochastic montotone submodular function subject to a cardinality constraint. Section 4 contains our main results for the problem of maximizing a stochastic monotone submodular welfare function subject to cardinality constraints. Section 5 presents a numerical evaluation of a test score algorithm for a simple illustrative example, a tight characterization of approximation guarantees achieved by mean and quantile test scores for the CES value function, and some discussion points. Finally, we conclude in Section 6. All the proofs of theorems and additional discussions are provided in Appendix.

2. Model and Problem Formulation

In this section, we introduce basic definitions for submodular functions, more formal definitions of the optimization problems that we study, and examples of various value functions.

2.1. Preliminaries: Submodular Functions

Given a ground set $N = \{1, 2, \ldots, n\}$ of items or elements with 2^N denoting the set of all possible subsets of N, a set function $u : 2^N \to \mathbf{R}_+$ is *submodular* if $u(S \cup T) + u(S \cap T) \leq u(S) + u(T)$, for all $S, T \in 2^N$. This condition is equivalent to saying that u satisfies the intuitive *diminishing returns property*: $u(T \cup \{i\}) - u(T) \leq u(S \cup \{i\}) - u(S)$ for all $i \in N \setminus T$ and $S, T \in 2^N$ such that $S \subseteq T$. Furthermore, we say that u is *monotone* if $u(S) \leq u(T)$ for all $S, T \in 2^N$ such that $S \subseteq T$.

We adopt the definition of a stochastic submodular function, e.g. used in (Asadpour and Nazerzadeh 2016), as the expected value of a submodular value function. Let $g : \mathbf{R}^n \to \mathbf{R}_+$ be a *value function* (Asadpour and Nazerzadeh 2016) that maps n-dimensional vectors to non-negative reals—g is said to be a *submodular value function* if for any two vectors $\mathbf{x}$ and $\mathbf{y}$ belonging to its domain:

$$g(\mathbf{x} \vee \mathbf{y}) + g(\mathbf{x} \wedge \mathbf{y}) \leq g(\mathbf{x}) + g(\mathbf{y}). \tag{3}$$

In the above definition, $\mathbf{x} \vee \mathbf{y}$ denotes the component-wise maximum and $\mathbf{x} \wedge \mathbf{y}$ the component-wise minimum. Note that when the domain of g is the set of Boolean vectors (all elements taking either value 0 or 1), then (3) reduces to the definition of a submodular set function. Hence, submodular value functions are a strict generalization of submodular set functions. Finally, we say that the value function g is monotone if for any two vectors $\mathbf{x}$ and $\mathbf{y}$ satisfying $\mathbf{y} \geq \mathbf{x}$ ($\mathbf{y}$ dominates $\mathbf{x}$ component-wise), we have $g(\mathbf{y}) \geq g(\mathbf{x})$.

For every $S \in 2^N$, we define $\mathbf{x} \mapsto M_S(\mathbf{x})$ to be a mapping such that $M_S(\mathbf{x})_i = x_i$ if $i \in S$ and $M_S(\mathbf{x}) = \phi$, otherwise. Here, ϕ is a minimal element which does not change the function value by adding an item of individual value ϕ. For example, for the value function $g(\mathbf{x}) = \max\{x_1, x_2, \ldots, x_n\}$ defined on $\mathbf{R}_+^n$, we may define $\phi = 0$. We are now ready to define stochastic submodular functions. Suppose that each item $i \in N$ is associated with a non-negative random variable X_i that is drawn independently from distribution P_i. We assume that each P_i is a cumulative distribution function, i.e. $P_i(x) = \mathbf{Pr}[X_i \leq x]$. Given a monotone submodular value function g, we define the stochastic monotone submodular function $u : 2^N \to \mathbf{R}_+$ by

$$u(S) = \mathbf{E}[g(M_S(X_1, X_2, \ldots, X_n))]. \tag{4}$$

For example, if g is the max or best-shot function, then $u(S) = \mathbf{E}[\max_{i \in S}\{X_i\}]$.

The following result, from Lemma 3 in Asadpour and Nazerzadeh (2016), establishes that under the prevailing assumptions, u is a submodular set function.

Lemma 1. *Suppose that g is a monotone submodular value function. Then, any set function u satisfying* (4) *is a monotone submodular set function.*

2.2. Problem Definitions

In this work, we study the design of test score algorithms for two combinatorial optimization problems, namely: (a) maximizing a stochastic monotone submodular function subject to a cardinality constraint, and (b) maximizing a stochastic monotone submodular welfare function defined as the sum of stochastic monotone submodular functions subject to cardinality constraints. We begin with the first problem. Recall the optimization problem presented in (1), and suppose that $\mathcal{F} = \{S \subseteq N \mid |S| = k\}$ for a given cardinality constraint $0 < k \leq n$ and let $\mathbf{X} = (X_1, \ldots, X_n)$ be a vector of random, independently distributed item performances such that $\mathbf{Pr}[X_i \leq x] = P_i(x)$.

By recasting problem (1) in terms of the notation developed in Section 2.1, we can now define the problem of maximizing a stochastic monotone submodular function subject to a cardinality constraint k as follows[2]

$$\arg\max_{S\in\mathcal{F}} u(S) := \mathbf{E}[f(S,\mathbf{X})] := \mathbf{E}[g(M_S(\mathbf{X}))], \tag{5}$$

where g is a monotone submodular value function. Additionally, we assume the function g to be *symmetric*, meaning that its value is invariant to permutations of its input arguments, i.e. for every $\mathbf{x}\in\mathbf{R}^n$, $g(\mathbf{x}) = g(\pi(\mathbf{x}))$ for any permutation $\pi(\mathbf{x})$ of the elements $\{x_1, x_2, \ldots, x_n\}$. The symmetric nature of the function is motivated by scenarios where the group value of a set of items depends on the individual performance values than the identity of the members who generate these values. For example, in the case of non-hierarchical team selection, it is reasonable to argue that two mutually exclusive teams S, T whose members yield identical performances on a given day also end up providing the same group value. Similarity, in content recommendation, the probability that user clicks on at least one topic can be viewed as a function of the user's propensity to click on each individual topic. Finally, by seeking to optimize the expected function value in (5), we implicitly model a risk-neutral decision maker as is typically the case in online platforms.

The stochastic submodular maximization problem specified in (5) is NP-hard even when the value function g is symmetric (in fact, Goel et al. (2006) show this is true for $g(\mathbf{x}) = \max\{x_1, \ldots, x_n\}$), and hence, we focus on finding approximation algorithms. Formally, given $\alpha \in [0,1]$, an algorithm is said to provide an α-approximation guarantee for (5) if for any given instance of the problem with optimum solution set OPT, the solution S returned by the algorithm satisfies $u(S) \geq \alpha u(\mathrm{OPT})$. Although a variety of approximation algorithms have been proposed for the submodular maximization problem, in this work, we focus on a special class of methods we refer to as *test score algorithms*. Specifically, these are algorithms that take as input a vector of non-negative test scores $a_1, a_2, \ldots, a_n$, and use only these scores to determine a feasible solution S for the problem (5). As defined in (2), the value of each test score a_i can depend only on g, k, and P_i. Furthermore, we are particularly interested in proposing test score algorithms that simply select the k items with the highest test scores in $\{a_1, a_2, \ldots, a_n\}$; such an approach is naturally appealing due to its intuitive interpretation. Clearly, the main challenge in this case is to design a suitable test score mapping rule that enables such a trivial algorithm to yield desirable guarantees.

[2] We used the formulation $u(S) = \mathbf{E}[f(S,\mathbf{X})]$ in the introduction to maintain consistency with the literature on stochastic optimization, e.g., (Kleywegt et al. 2002). For the rest of this paper, we will exclusively write $u(S) = \mathbf{E}[g(M_S(\mathbf{X}))]$ for convenience and to delineate the interplay between the set S and the submodular value function g.

Stochastic Submodular Welfare Maximization Maximizing a stochastic submodular welfare function is a strict generalization of the problem of maximizing a stochastic monotone submodular function subject to a cardinality constraint as defined in (5). Here, we are given a ground set $N = \{1, \dots, n\}$, and a collection of stochastic monotone submodular set functions $u_j : 2^N \to \mathbf{R}_+$ with corresponding submodular value functions $g_j : \mathbf{R}^n \to \mathbf{R}_+$ for $j \in M := \{1, 2, \dots, m\}$. The goal is to find disjoint subsets $S_1, S_2, \dots, S_m$ of the ground set of given cardinalities $|S_1| = k_1$, $|S_2| = k_2$, $\dots$, $|S_m| = k_m$ that maximize the welfare function defined as

$$u(S_1, S_2, \dots, S_m) = \sum_{j=1}^{m} u_j(S_j). \tag{6}$$

We refer to M as the set of partitions. Similarly as for the previous problem, we consider symmetric, monotone, submodular value functions g_j for each partition $j \in M$ so that the stochastic submodular set functions can be represented as follows:

$$u_j(S) = \mathbf{E}[g_j(M_S(X_{1,j}, X_{2,j}, \dots, X_{n,j}))] \quad \text{for all } j \in M.$$

In the above expression, $X_{i,j}$ denotes the individual performance of item $i \in N$ with respect to partition $j \in M$. Each $X_{i,j}$ is drawn independently from a distribution $P_{i,j}$ that is the cumulative distribution function $P_{i,j}(x) = \mathbf{Pr}[X_{i,j} \leq x]$. Our formulation allows for considerable heterogeneity as items can have different realizations of their individual performances for different partitions. Submodular welfare maximization problems arise naturally in domains such as team formation where decision makers are faced with the problem of selecting agents and assigning them to projects or teams. For example, this could model an online gaming platform seeking to choose a collection of teams to participate in a tournament or an organization partitioning its employees to focus on asymmetric tasks. In these situations, the objective function (6) captures the aggregate value generated by all of the teams.

Once again, we are interested in designing test score algorithms for stochastic submodular welfare maximization. More formally, a test score algorithm for problem (6) is a procedure whose input only comprises of vectors of test scores $(\mathbf{a}_{i,j})_{i \in N, j \in M}$, where the elements of each test score vector $\mathbf{a}_{i,j}$ are a function of g_j, k_j, and $P_{i,j}$. Note that in this general formulation, each item $i \in N$ and partition $j \in M$ is associated with vector-valued test scores $\mathbf{a}_{i,j}$.

2.3. Examples of Value Functions

Many functions used in the literature to model both online as well as traditional production functions belong to the class of symmetric, stochastic monotone submodular value functions. In this section, we introduce and discuss several well known examples.

A common value function is defined to be an increasing function of the sum of individual values: $g(\mathbf{x}) = \bar{g}\left(\sum_{i=1}^{n} x_i\right)$, where $\bar{g}$ is a non-decreasing concave function and thus exhibits diminishing returns. Ahmed and Atamtürk (2011) illustrate how these value functions can be utilized to model diverse applications—e.g., risk averse capital budgeting under uncertainty, and competitive facility location. One specific example of such a function is the threshold function, i.e., $g(\mathbf{x}) = \min\{\sum_{i=1}^{n} x_i, B\}$ for some $B > 0$, which captures the behavior of budget-constrained buyers in markets, most prominently in online advertising (Mehta et al. 2013).

The *best-shot* or the *max function* is commonly used to aggregate multiple inputs and is defined as $g(\mathbf{x}) = \max\{x_1, x_2, \ldots, x_n\}$. This value function allows us to model scenarios when only the best individual output contributes to group value. For example, this arises in online crowdsourcing systems in which solutions to a problem are solicited by an open call to the online community, several candidate solutions are received, but eventually only the best one is used.

A natural generalization of the best-shot function is a *top-r value function* defined as the sum of r highest individual values, for a given parameter $1 \leq r \leq n$, i.e. $g(\mathbf{x}) = x_{(1)} + x_{(2)} + \cdots + x_{(r)}$, where $x_{(i)}$ is the i-th largest element of input vector $\mathbf{x}$. This function reduces to the best-shot value function when $r = 1$. The top-r value function is of interest in a variety of applications such as information retrieval and recommender systems, where the goal is to identify a set of most relevant items.

A well known value function is the *constant elasticity of substitution (CES) function*, which is defined by $g(\mathbf{x}) = \left(\sum_{i=1}^{n} x_i^r\right)^{1/r}$, for a positive value parameter $r \geq 1$. This value function has been in common use to model production systems in economics and team selection (Fu et al. 2016, Dixit and Stiglitz 1977). The family of CES value functions accommodates different types of production by suitable choices of parameter r, including the linear production for $r = 1$ and the best-shot production in the limit as the value of r goes to infinity.

Finally, we make note of the *success probability value function*, defined by $g(\mathbf{x}) = 1 - \prod_{i=1}^{n}(1 - p(x_i))$, where $p : \mathbf{R} \rightarrow [0, 1]$ is an increasing function satisfying $p(0) = 0$. This value function is often used as a model of tasks for which input solutions are independent and either good or bad (success or failure), and it suffices to have at least one good solution for the task to be successfully solved, e.g., see Kleinberg and Oren (2011).

2.4. Computation, Implementation, and the Role of Value Oracles

We conclude this section with a discussion of some practical issues surrounding test scores algorithms and function evaluation. Given that submodular set functions have representations that are exponential in size (2^n), a typical modeling concession is to assume access to a value oracle for function evaluation. Informally, a value oracle is a black-box that when queried with any set

$S \in 2^N$, returns the function value $u(S)$ in constant time. Although value oracles are a theoretically convenient abstraction, function evaluation can be rather expensive in applications pertaining to online platforms. This problem is further compounded in the case of stochastic submodular functions when the underlying item performance distributions $(P_1, P_2, \ldots, P_n)$ are unknown. Naturally, one would expect a non-zero query-cost to be associated with evaluating $g(\mathbf{x})$ even for a single realization $\mathbf{x}$ of the random vector $\mathbf{X}$. Under these circumstances, there is a critical need for algorithms that achieve desirable guarantees using significantly fewer queries and to eschew traditional approaches (e.g., greedy) that require polynomially many oracle calls.

To illustrate these challenges, consider the content recommendation application from Example 1 and suppose that both the distributions $P_1, P_2, \ldots, P_n$ and the value function g are unknown. In order to (approximately) compute $u(S)$ for any $S \subseteq N$, it is necessary to present the set S of topics repeatedly to a large number of users and average their response (e.g., upvotes or click behavior). Clearly, a protracted experimentation phase brought about by too many oracle queries could lead to customer dissatisfaction or even a loss in revenue. Alternatively, in team hiring or online gaming, evaluating the function value for arbitrary subsets $S \subseteq N$ may be prohibitively expensive as it may not be possible to observe group performance before the team is even formed. The replication test score algorithm proposed in Section 3 addresses these issues by not only making use of fewer oracle calls but also allowing for easier implementation since each evaluation of the function g only requires samples from a single item's (or agent's) performance distribution P_i.

A secondary issue concerns the noise in the function evaluation or test score computation brought about by sampling from the distributions $P_1, P_2, \ldots, P_n$. It may not be possible to precisely compute test scores a_i that represent the expected value of some function under distribution P_i—e.g., mean test scores where $a_i = \mathbf{E}_{X_i \sim P_i}[X_i]$ or replication test scores in (7). In practice, test scores could be defined as sample estimators with values determined by the observed data, i.e., utilize a sample mean instead of the population mean. In our analysis, we ignore the issue of estimation noise and assume oracle access that facilitates the precise computation of test scores that denote some expectation with respect to distributions $P_1, P_2, \ldots, P_n$. This assumption is justified provided that the estimators are unbiased and the test scores are estimated using a sufficient number of samples. We leave accounting for statistical estimation noise as an item for future research.

Finally, it is worth highlighting that the benefits of test score algorithms do not come without a price. Using a test score based approach severely limits what an algorithm can do, which in turn may affect the achievable quality of approximation. For instance, the aforementioned greedy algorithm is able to leverage its unrestricted access to a value oracle and achieve a $1 - 1/e$-approximation for (5) by carefully querying the function value for many different subsets $S \in 2^N$. Test score algorithms, however, do not have this luxury—instead, they rely indirectly on approximating answers to value oracle queries using only limited information, namely parameters associated with individual items $i \in N$ evaluated separately on the function g.

3. Submodular Function Maximization

In this section we present our main result on the existence of test scores that guarantee a constant-factor approximation for maximizing a stochastic monotone submodular function subject to a cardinality constraint, for symmetric submodular value functions that satisfy an extended diminishing returns condition. We begin by introducing some basic terminology required for our sufficient condition. Given a value function $g : \mathbf{R}_+^n \to \mathbf{R}_+$ and $v \geq 0$, we say that v has a non-empty *preimage* under g if there exists at least one $\mathbf{z} \in \mathbf{R}_+^n$ such that $g(\mathbf{z}) = v$. For simplicity, we slightly abuse notation and ignore the ϕ-elements when providing the inputs to function g—e.g., given $\mathbf{x} = (x_1, x_2, \dots, x_d, \phi, \dots, \phi) \in \mathbf{R}_+^n$ and $\tilde{\mathbf{x}} = (x_1, x_2, \dots, x_d)$, we write $g(\tilde{\mathbf{x}})$ instead of $g(\mathbf{x})$.

Definition 1 (Extended Diminishing Returns Condition). A symmetric submodular value function $g : \mathbf{R}_+^n \to \mathbf{R}_+$ is said to satisfy the *extended diminishing returns condition* if for every $v \geq 0$ that has a non-empty preimage under g, there exists $\mathbf{z} \in \mathbf{R}_+^{n-1}$ such that:

(a) $g(\mathbf{z}) = v$, and

(b) for all $\mathbf{y} \in \mathbf{R}_+^{n-1}$ such that $g(\mathbf{y}) \leq v$, $g(\mathbf{y}, x) - g(\mathbf{y}) \geq g(\mathbf{z}, x) - g(\mathbf{z})$ for all $x \in \mathbf{R}_+$.

Informally, the condition states that given that a value v such that the function evaluates to this number at one or more points in its domain, then for at least one such point, say $\mathbf{z}$, the marginal benefit of adding an element of value x to $\mathbf{z}$ cannot be larger than the marginal benefit of adding the same element to another vector $\mathbf{y}$ whose performance is smaller than v. The extended submodularity condition holds for a wide range of functions. For example, the condition is satisfied by all value functions defined and discussed in Section 2.3, which is proved in Appendix A.

We refer to this property as extended diminishing returns as it is consistent with the spirit of 'decreasing marginal returns' as the function value grows. Indeed, as in the case of traditional submodular functions, adding a new element (x) provides greater marginal benefit to a vector yielding a smaller performance ($\mathbf{y}$) than to one providing a larger value ($\mathbf{z}$). In other words, we have diminishing marginal returns as the value provided by a vector $\mathbf{z}$ grows. Consider for example, a team application: a new member with some potential would be expected to make a less significant contribution to a high performing team than a low performing one. Similarly, in content recommendation, the added benefit provided by a new topic would be felt more strongly by a user who derives limited value from the original assortment than one who was highly satisfied to begin with. The underlying mechanism in both these examples is that a new member or topic would have a greater overlap in skills or content with a high performing group of items.

A subtle point is worth mentioning here. For any given v, if there exist multiple points in the domain at which the function g evaluates to v, then the extended diminishing returns condition only guarantees the existence of a single vector $\mathbf{z}$ for which $g(\mathbf{z},x) - g(\mathbf{z}) \geq g(\mathbf{y},x) - g(\mathbf{y})$ holds for all $\mathbf{y},x$ such that $g(\mathbf{y}) \leq v$. Simply put, there may be other vectors which also evaluate to v which do not satisfy the above inequality.[3] We remark that this is actually a weaker requirement than imposing that all such vectors satisfy condition (b) in Definition 1—this allows our results to be applicable to a broader set of functions. That being said, most of the value functions that we specify in Section 2.3 except for top-r $(r > 1)$ satisfy a stronger version of extended diminishing returns where the condition $g(\mathbf{z},x) - g(\mathbf{z}) \geq g(\mathbf{y},x) - g(\mathbf{y})$ holds for every two points $\mathbf{z},\mathbf{y} \in \mathbf{R}_+^{n-1}$ such that $g(\mathbf{y}) \leq g(\mathbf{z})$.

We next introduce the special type of test scores, we refer to as replication test scores.

Definition 2 (replication test scores). Given a symmetric submodular value function g and cardinality parameter k, for every item $i \in N$, the replication test score a_i is defined by

$$a_i = \mathbf{E}[g(X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(k)}, \phi, \ldots, \phi)] \tag{7}$$

where $X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(k)}$ are independent and identically distributed random variables with distribution P_i.

The replication test score of an item can be interpreted as the expected performance of a *virtual* group of items that consists of k independent replicas of this item, hence the name replication scores. Note that a replication test score is defined for a given function g and cardinality parameter k; we omit to indicate this in the notation a_i for simplicity.

In contrast to mean or quantile test scores that simply provide some measure of an item's performance, replication test scores capture both the item's individual merit as well as its contribution to a group of items. To understand this distinction, consider Example 1 where $g(\mathbf{x}) = \max\{x_1, x_2, \ldots, x_n\}$ and $p = 1/k$. Clearly, the mean performance of stable type A items (a) is larger than the mean performance of polarizing topics of type B (b). However, the replication score of a type B item is $(1-(1-p)^k)\frac{b}{p} \geq (1-\frac{1}{e})bk$ which for large enough k is larger than the replication score of a type A item which still remains a. The larger replication score of type B topics captures the intuition that risky topics can often provide significant marginal benefits to an existing assortment. Finally, in the case of content recommendation, one can employ a natural mechanism to estimate the replication scores even when the objective function g and distributions $P_1, P_2, \ldots, P_n$ are unknown. Namely, in order to compute the replication score for a topic of type A (or B), it suffices to present k items of this type to a large number of incoming users and compute the average response.

We next present the main result of this section.

[3] Suppose that g is the top$-r$ function defined in Section 2.3 for $r = 2$, $\mathbf{x} = (1,1)$, and $v = 4$. Consider vectors $\mathbf{y}_1 = (2,2)$ and $\mathbf{y}_2 = (4)$ such that $g(\mathbf{y}_1) = g(\mathbf{y}_2) = v = 4$. It is not hard to deduce that for any $0 < z \leq 1$, $g(\mathbf{x},z) - g(\mathbf{x}) = 0 = g(\mathbf{y}_1,z) - g(\mathbf{y}_1) < g(\mathbf{y}_2,z) - g(\mathbf{y}_2) = z$. That is $\mathbf{y}_1$ satisfies the conditions in Definition 1 but $\mathbf{y}_2$ does not.

Theorem 2. *Suppose that the utility set function is the expected value of a symmetric, monotone submodular value function that satisfies the extended diminishing returns condition. Then, the greedy selection of items in decreasing order of replication test scores yields the utility value that is at least $(1-1/e)/(5-1/e)$ times the optimal value.*

In the remainder of this section, we prove Theorem 2. Along the way, we derive several results that connect the underlying discrete optimization problem with approximating set functions, which may be of independent interest.

The key mathematical concept that we use is a *sketch* of a set function, which is an approximation of a potentially complicated set function using simple polynomial-time computable lower and upper bound set functions, we refer to as a *minorant* and a *majorant* sketch function, respectively. Majorants and minorants are functions that serve as upper and lower bounds respectively for the given set function at all inputs.

Definition 3 (Sketch). A pair of set functions $(\underline{v}, \bar{v})$ is said to be a (p,q)-sketch of a set function $u : 2^N \rightarrow \mathbf{R}_+$, if the following condition holds:

$$p\underline{v}(S) \leq u(S) \leq q\bar{v}(S), \text{ for all } S \subseteq N. \tag{8}$$

The set functions $\underline{v}$ and $\bar{v}$ are referred to as a *minorant* and a *majorant* sketch function, respectively. In particular, if (v,v) is a (p,q)-sketch, we refer to v as a *strong sketch* function.[4]

Although the above definition is quite general, and subsumes many trivial sketches (for e.g, $\underline{v}=0, \bar{v}=\infty$), practically useful sketches would satisfy a few fundamental properties such as (a) when given a set function whose description may be exponential in n, $\underline{v}$ and $\bar{v}$ must be polynomially expressible, and (b) $\underline{v}$ and $\bar{v}$ must be sufficiently close to each other at points of interest for the sketch to be meaningful. Our first result provides sufficient conditions on the sketch functions to obtain an approximation algorithm for maximizing a monotone submodular set function subject to a cardinality constraint.

Lemma 3. *Suppose that (a) $\underline{v}$ and $\bar{v}$ are minorant and majorant set functions that are a (p,q)-sketch of a submodular set function $u : 2^N \rightarrow \mathbf{R}_+$ and (b) there exists $S^* \subseteq \arg\max_{S:|S|=k} \underline{v}(S)$ that satisfies $\bar{v}(S) \leq \underline{v}(S^*)$ for every $S \subseteq N$ that has cardinality k and is completely disjoint from S^*, i.e. $S \cap S^* = \emptyset$. Then, the following relation holds:*

$$u(S^*) \geq \frac{p}{q+p} u(\mathrm{OPT}),$$

where OPT *denotes an optimum set of cardinality k.*

[4] Our definition of a strong sketch is closely related to the following definition of a sketch used in literature (e.g., see Cohavi and Dobzinski (2017)): a set function $\tilde{v}$ is said to be a α-sketch of u if $\tilde{v}(S) \leq u(S) \leq \alpha\tilde{v}(S)$ for all $S \subseteq N$. Indeed, if v is a (p,q)-strong sketch of u, then $\tilde{v}(S) := pv(S)$ is a q/p-sketch of u.

The proof of Lemma 3 is provided in Appendix B. The proofs follows by basic properties of submodular set functions and conditions of the lemma.

The result in Lemma 3 tells us that if we can find a minorant set function $\underline{v}$ and a majorant set function $\bar{v}$ that are a (p,q)-sketch for a submodular set function u and that satisfy the conditions of the lemma, then any solution of the problem of maximizing the function $\underline{v}$ subject to a cardinality constraint is a $p/(p+q)$-approximation for the problem of maximizing the submodular set function u subject to the same cardinality constraint. What remains to be done is to find such minorant and majorant set functions, and show that for every set $S \subseteq N$, the value of these functions can be computed in polynomial-time by using only test scores of items in S.

We define a minorant set function $\underline{v}$ and a majorant set function $\bar{v}$, for any given test scores $a_1, a_2, \dots, a_n$, as, for $S \subseteq N$,

$$\underline{v}(S) = \min\{a_i \mid i \in S\} \text{ and } \bar{v}(S) = \max\{a_i \mid i \in S\}. \tag{9}$$

For the minorant set function $\underline{v}$ defined in (9), the problem of maximizing $\underline{v}(S)$ over $S \subseteq N$ subject to cardinality constraint $|S| = k$ corresponds to selecting a set of k items with largest test scores. Obviously, the set functions $\underline{v}$ and $\bar{v}$ defined in (9) satisfy condition (b) in Lemma 3.

We only need to show that there exist test scores $a_1, a_2, \dots, a_n$ such that $(\underline{v}, \bar{v})$ is a (p,q)-sketch of the set function u. We say that $a_1, a_2, \dots, a_n$ are (p,q)-good test scores if $(\underline{v}, \bar{v})$ is a (p,q)-sketch of the set function u. If p/q is a constant, we refer to $a_1, a_2, \dots, a_n$ as *good test scores*. In this case, by Lemma 3, selecting a set of k items with largest test scores guarantees a constant-factor approximation for the problem of maximizing the set function $u(S)$ subject to the cardinality constraint $|S| = k$. More generally, we have the following corollary.

Corollary 4. *Suppose that test scores $a_1, a_2, \dots, a_n$ are (p,q)-good. Then, greedy selection of items in decreasing order of these test scores yields a utility of value that is at least $p/(p+q)$ times the optimum value. Moreover, if p/q is a constant, than the greedy selection guarantees a constant-factor approximation for maximizing the submodular set function $u(S)$ subject to the cardinality constraint $|S| = k$.*

We next need to address the question whether for a given stochastic monotone submodular function there exist good test scores. If good test scores exist, it is possible that there are different definitions of test scores that are good test scores. The following lemma shows that replication test scores, defined in Definition 2, are good test scores, whenever good test scores exist.

Lemma 5. *Suppose that a utility function has (p,q)-good test scores. Then, for this utility function, the replication scores are $(p/q, q/p)$-good test scores.*

The proof of Lemma 5 is provided in Appendix C. The lemma tells us that to check whether a utility function has good test scores, it suffices to check whether for this utility function, replication test scores are good test scores. If replication test scores are not good test scores for a given utility function, then there exist no good test scores for this utility function.

In the next lemma, we show that the extended diminishing returns condition, which we introduced in Definition 1, is a sufficient condition for replication test scores to be good test scores.

Lemma 6. *Suppose that $g:\mathbf{R}_+^n \to \mathbf{R}_+$ is a symmetric, monotone submodular value function that satisfies the extended diminishing returns condition. Then, the replication test scores are $(1-1/e, 4)$-good test scores, and consequently are good test scores.*

The proof of Lemma 6 is provided in Appendix D. Here we briefly discuss some of the main ideas of the proof. First, for the lower bound, we need to show that for every set $S \subseteq N$: $u(S) \geq (1-1/e)\underline{v}(S) = (1-1/e)\min\{a_i \mid i \in S\}$, where a_i is the replication test score of item i. Suppose that $S = \{1,2,\dots,k\}$ and without loss of generality, $a_1 = \min\{a_i \mid i \in S\}$. Then, we show by induction that for every $j \in \{1,\dots,k\}$,

$$u(\{1,2,\dots,j\}) \geq \left(1-\frac{1}{k}\right)u(\{1,2,\dots,j-1\}) + \frac{1}{k}a_1. \tag{10}$$

The proof involves showing that the marginal contribution of adding item j to the set $\{1,2,\dots,j-1\}$ is closely tied to the marginal contribution of adding item j to a set comprising of $k-1$ other (independently drawn) copies of item j. The latter quantity is at most a_j/k, which by definition is greater than or equal to a_1/k. The exact factor of $1-1/e$ comes from applying the above inequality in a cascading fashion from $u(\{1,2,\dots,k\})$ to $u(\{1\})$.

The proof of the upper bound is somewhat more intricate. The first step involves carefully constructing a vector $\mathbf{z} \in \mathbf{R}_+^{n-1}$ such that $g(\mathbf{z})$ is larger than $u(S)$ by an appropriate constant factor (say c) for a given S. Imagine that S^* represents some set of $n-1$ items such that $u(S^*) = g(\mathbf{z})$. By leveraging monotonicity and submodularity, we have that $u(S) \leq u(S^*) + \sum_{i\in S}(u(S^* \cup \{i\}) - u(S^*))$. Let $\mathbf{x}$ represent a vector comprising of $k-1$ independent copies of random variables drawn from distribution P_i. Now, as per the extended diminishing returns condition, for any realization of $\mathbf{x}$ such that $g(\mathbf{x}) \leq g(\mathbf{z})$, it must be true (assuming that the careful construction $\mathbf{z}$ leverages Definition 1) that:

$$u(S^* \cup \{i\}) - u(S^*)) = g(\mathbf{z}, x_i) - g(\mathbf{z}) \leq g(\mathbf{x}, x_i) - g(\mathbf{x}) \text{ given that } g(\mathbf{x}) \leq g(\mathbf{z}).$$

Moreover, one can apply Markov's inequality to show that $g(\mathbf{z}) \geq g(\mathbf{x})$ is true with probability at least $1/c$. Taking the expectation of $\mathbf{x}$ conditional upon $g(\mathbf{z}) \geq g(\mathbf{x})$ gives us the desired upper bound. The statement of Theorem 2 follows from Corollary 4 and Lemma 6.

4. Submodular Welfare Maximization

In this section, we present our main result for the stochastic submodular welfare maximization problem. Here, the goal is to find disjoint sets $S_1, S_2, \ldots, S_m \subseteq N$ satisfying cardinality constraints $|S_j| = k_j$ for $j \in \{1, 2, \ldots, m\}$ that maximize the welfare function $u(S_1, S_2, \ldots, S_m) = \sum_{j=1}^{m} u_j(S_j)$. The main theorem for the stochastic submodular welfare maximization problem is stated as follows.

Theorem 7. *Given an instance of the submodular welfare maximization problem such that the utility functions satisfy the extended diminishing returns condition, and the maximum cardinality constraint is k, there exists a test score algorithm (Algorithm 1) that achieves a welfare value of at least $1/(24(\log(k)+1))$ times the optimum value.*

We briefly comment on the efficacy of test score algorithms for the submodular welfare maximization problem. Unlike the constant factor approximation guarantee obtained in Theorem 2, test score algorithms only yield a logarithmic-approximation to the optimum for this more general problem. Although constant factor approximation algorithms are known for the submodular welfare maximization problem (Calinescu et al. 2011), these approaches rely on linear programming and other complex techniques and hence, may not be scalable or amenable to distributed implementation. On the other hand, we focus on an algorithm that is easy to implement in practice but relies on a more restrictive computational model, leading to a worse approximation. Finally, it is worth noting that in many actual settings, the value of the cardinality constraint k is typically much smaller in comparison to n; e.g., in content recommendation, it is typical to display 20-25 items per page presented to a user while the total number of items is much lager. In such cases, the loss in approximation due to the logarithmic factor would not be significant.

In the remainder of this section, we provide a proof of Theorem 7. We will present an algorithm that uses replication test scores, which achieves the logarithmic guarantee. The proof is based on using strong sketches of set functions.

We follow the same general framework as for the submodular function maximization problem, presented in Section 3, which in this case amounts to identifying a strong sketch function for each utility set function, defined by using replication test scores, and then using a greedy algorithm for welfare maximization that carefully leverages these replication test scores to achieve the desired approximation guarantee. The following lemma establishes a connection between the submodular welfare maximization problem and strong sketches.

Lemma 8. *Consider an instance of the submodular welfare maximization with utility functions $u_1, u_2, \ldots, u_m$ and parameters of the cardinality constraints $k_1, k_2, \ldots, k_m$. Let $\mathrm{OPT} = (\mathrm{OPT}_1, \mathrm{OPT}_2, \ldots, \mathrm{OPT}_m)$ denote an optimum partition of items. Suppose that for each $j \in M$,*

(v_j, v_j) is a (p,q)-sketch of u_j, and that $S_1, S_2, \ldots, S_m$ is an α-approximation to the welfare maximization problem with utility functions $v_1, v_2, \ldots, v_m$ and the same cardinality constraints. Then,

$$\sum_{j=1}^{m} u_j(S_j) \geq \alpha \frac{p}{q} u(\mathrm{OPT}_1, \ldots, \mathrm{OPT}_m) = \alpha \frac{p}{q} \sum_{j=1}^{m} u_j(\mathrm{OPT}_j).$$

The proof of Lemma 8 is provided in Appendix E.

We next define set functions that we will show to be a strong sketch for utility functions of the welfare maximization problem that satisfy the extended diminishing returns condition. Fix an arbitrary set $S \subseteq N$ such that $|S| = k$ and $j \in M$. Let $a_{i,j}^r$ be the replication score of item i for value function g_j and cardinality parameter r, i.e.,

$$a_{i,j}^r = \mathbf{E}[g_j(X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(r)}, \phi, \ldots, \phi)].$$

Let $\pi(S,j) = (\pi_1(S,j), \ldots, \pi_k(S,j))$ be a permutation of items in S defined as follows:

$$\begin{aligned} \pi_1(S,j) &= \arg\max\nolimits_{i \in S} a_{i,j}^1 \\ \pi_2(S,j) &= \arg\max\nolimits_{i \in S \setminus \{\pi_1(S,j)\}} a_{i,j}^2 \\ &\vdots \\ \pi_k(S,j) &= \arg\max\nolimits_{i \in S \setminus \{\pi_1(S,j), \ldots, \pi_{k-1}(S,j)\}} a_{i,j}^k. \end{aligned} \tag{11}$$

We define a set function $v_j : 2^N \to \mathbf{R}_+^n$ for every set $S \subseteq N$ of cardinality k as follows:

$$v_j(S) = a_{\pi_1(S,j),j}^1 + \frac{1}{2} a_{\pi_2(S,j),j}^2 + \cdots + \frac{1}{k} a_{\pi_k(S,j),j}^k. \tag{12}$$

The definition of set function v_j in (12) can be interpreted as defining the value $v_j(S)$ for every given set S to be additive with coefficients associated with items corresponding to their virtual marginal values in a greedy ordering of items with respect to these virtual marginal values.

Given a partition of items in disjoint sets $S_1, S_2, \ldots, S_m$, we define a welfare function $v(S_1, S_2, \ldots, S_m) = \sum_{j=1}^m v_j(S_j)$. We next show that the set functions defined in (12) are strong sketch functions.

Lemma 9. *Suppose that a set function u_j is defined as the expected value of a symmetric, monotone submodular value function that satisfies the extended diminishing returns condition. Then, the set function v_j given by (12) is a $(1/(2(\log(k)+1)), 6)$ strong sketch of u_j.*

The proof of Lemma 9 is provided in Appendix F.

We refer to the problem of maximizing the welfare function $v(S_1, S_2, \ldots, S_m)$ subject to the cardinality constraints as the *surrogate welfare maximization problem.* By Lemma 8 and Lemma 9, for any stochastic monotone submodular welfare maximization problem with utility functions

ALGORITHM 1: Greedy Algorithm for Submodular Welfare Maximization Problem

```
Initialize assignment $S_1 = S_2 = \ldots = S_m = \emptyset$  $A = \{1, 2, \ldots, n\}$, $P = \{1, 2, \ldots, m\}$
/* $S_j$ and $A$ denote the set of assigned items to partition $j$ and the set of unassigned items */
while $|A| > 0$ and $|P| > 0$ do
    $(i^*, j^*) = \arg\max_{(i,j) \in A \times P} a_{i,j}^{|S_j|+1} / (|S_j| + 1)$    /* with random tie break */
    $S_{j^*} \leftarrow S_{j^*} \cup \{i^*\}$ and $A \leftarrow A \setminus \{i^*\}$    /* assign item $i^*$ to partition $j^*$ */
    if $|S_{j^*}| \geq k_j$ then
        $P \leftarrow P \setminus \{j^*\}$    /* remove partition $j^*$ from the list */
    end
end
```

satisfying the extended diminishing returns condition, any α-approximate solution to the surrogate welfare maximization problem is a $c\alpha/(\log(k) + 1)$-approximate solution to the original welfare maximization problem, where c is a positive constant. It remains to show that the surrogate welfare maximization problem admits an α-approximate solution. We next show that a natural greedy algorithms applied to the surrogate welfare maximization problem guarantees a 1/2-approximation for this problem.

Consider a natural greedy algorithm for the surrogate welfare maximization problem for the case of one or more partitions. Given the replication test scores for all items and all partitions, in each step r, the algorithm adds an unassigned item i and partition j that maximizes $a_{i,j}^{r_j}$ where r_j is the number of elements assigned to partition j in previous steps. That is, in each iteration, an assignment of an item to a partition is made that yields the largest marginal increment of the surrogate welfare function. The algorithm is more formally defined in Algorithm 1.

In the following lemma, we show that the greedy algorithm guarantees a 1/2-approximation for the surrogate welfare maximization problem.

Lemma 10. *The greedy algorithm defined by Algorithm 1 outputs a solution that is a $\frac{1}{2}$-approximation for the submodular welfare maximization problem of maximizing $v(S_1, S_2, \ldots, S_m)$ over partitions of items $(S_1, S_2, \ldots, S_m)$ that satisfy cardinality constraints.*

The proof of Lemma 10 can be found in Appendix 1. The proof is similar in spirit to that of the $\frac{1}{2}$-approximate greedy algorithm for submodular welfare maximization proposed by Lehmann et al. (2006). Unfortunately, one cannot directly utilize the arguments in that paper since the sketch function that we seek to optimize—$v_j(S_j)$—may not be submodular. Instead, we present a novel montonicity argument and leverage it to provide the following bounds: $v_j(S_j) \geq v_j(S_j \setminus \{\pi_r(S_j, j)\}) \geq v_j(S_j) - \frac{a^r_{\pi_r(S_j,j),j}}{|r|}$ for all $S_j \subseteq N$ and $1 \leq r \leq k_j$. Finally, we apply these bounds in a cascading manner to show the desired $\frac{1}{2}$-approximation factor claimed in Lemma 10.

5. Discussion and Additional Results

In this section, we first illustrate the use of test scores and discuss numerical results for the simple example introduced in Section 1. We then compare the performance of test score algorithms to the popular sample average approximation (SAA) approach for this example. Finally, we discuss the performance of simple scoring rules, namely mean and quantile scores, and characterize their performance for the constant elasticity of substitution (CES) value function. We focus on this function due to its extensive applications in other domains as highlighted earlier.

5.1. Numerical Results for a Simple Illustrative Example

We consider the example of two types of items that we introduced in Section 1. Recall, in this example the ground set of items is partitioned in two sets A and B with set A comprising of safe items whose individual performance is of value a with probability 1 and set B comprising of risky items whose individual performance is of value b/p with probability p, and value 0, otherwise. Here a, b, and p are parameters such that $a,b>0$ and $p\in(0,1]$. We assume that $b\geq a$ and $|A|,|B|\geq k$. The value function is assumed to be the best-shot (max) value function.

We say that a set S of items of cardinality k is of type r if it contains exactly r risky items for $r=0,1,\ldots,k$. For each $r\in\{0,1\ldots,k\}$, let S_r denote an arbitrary type-r set. The realized value of set S_r is b/p if at least one risky item in S_r achieves value b/p and is equal to a, otherwise. Hence, we have

$$u(S_r)=a(1-p)^r+\frac{b}{p}(1-(1-p)^r).$$

Notice that the value of $u(S_r)$ monotonically increases in r, hence it is optimal to select a set that comprises of k risky items, i.e. a set of type k.

In line with the practical implementation mentioned in Section 2.4, we consider an algorithm that estimates test scores by repeated sampling and selects the items with the highest sample-average replication scores. For a given number of samples per item replica $T\geq 1$, this is defined as:

$$\hat{a}_i=\frac{1}{T}\sum_{t=1}^{T}\max\{X_i^{(t,1)},X_i^{(t,2)},\ldots,X_i^{(t,k)}\}$$

where $X_i^{(t,j)}$ are independent samples over i, t and j with $X_i^{(t,j)}$ sampled from distribution P_i. Since the optimum set consists only of risky items, the output of the test score algorithm results in an error if, and only if, it contains at least one safe item. This occurs if the sample-average replication score of a safe item exceeds that of a risky item. We now evaluate the probability of error of the algorithm by running the test score algorithm for a number of repeated experiments.

In Figure 1, we show the probability of error versus the number of samples per item, for different values of parameters k and p. Notice that the total number of samples per item is equal to Tk

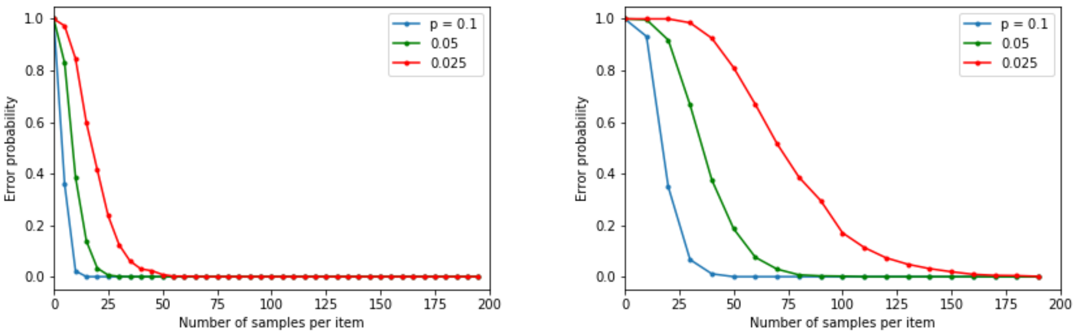

Figure 1 Probability of error of the test score algorithm versus the number of samples per item for (left) $k=5$ and (b) $k=10$, in each case for different values of parameter $p=0.025, 0.05$ and 0.1. Other parameters are set as $|A|=|B|=10$, $a=1$ and $b=2$. The results are averaged over 1000 repeated experiments.

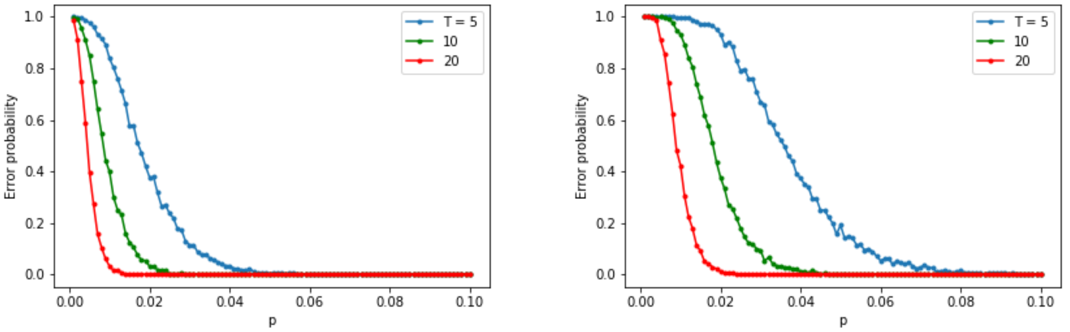

Figure 2 Probability of error of the test score algorithm versus the value of parameter p for (left) $k=5$ and (right) $k=10$, in each case for different number of samples per item replica $T=5$, 10 and 20. Other parameters are set as given in the caption of Figure 1.

where T is the number of samples per item replica. We observe that (a) the probability of error decreases with the number of samples per item, (b) the probability of error is larger for larger set size, and (c) the number of samples per item required to achieve a fixed value of probability of error increases with the risk of item values, i.e. for smaller values of parameter p. In Figure 2, we show the probability of error versus the value of the risk parameter p, for different values of parameters k and T. This further illustrates that a larger number of samples is needed to achieve a given probability of error as p decreases. In fact, one can prove that drawing at least $O((k/p^2)\log(n/\delta))$ samples per item is sufficient to guarantee that the probability of error does not exceed δ; we provide the corresponding details in Appendix H.

We further consider a sample averaging approach (SAA) that amounts to enumerating feasible sets of items, for each feasible set S of items estimating the value of $u(S)$, and selecting a set with

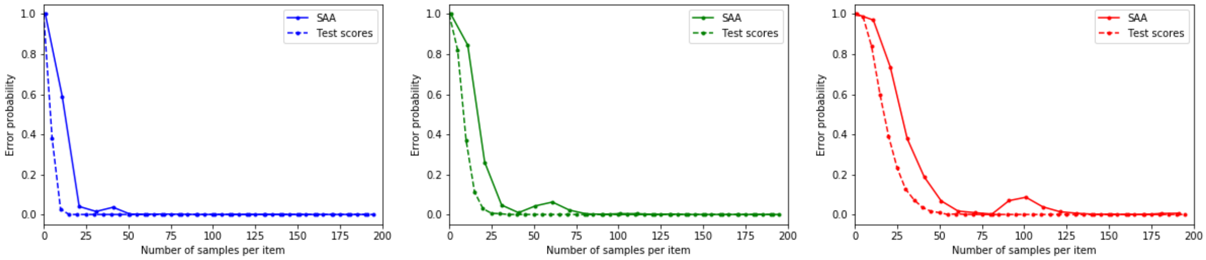

Figure 3 Probability of error versus the number of samples per item for SAA and test score algorithms, for (left) $p=0.1$, (middle) $p=0.05$, and (right) $p=0.025$. The setting of other parameters is as in Figure 1 for $k=5$.

the largest estimated value. The value of $u(S)$ is computed using the estimator defined as

$$\hat{u}(S) = \frac{1}{T}\sum_{t=1}^{T} \max\{X_i^{(t)} \mid i \in S\}$$

where $X_i^{(t)}$ are independent samples over i and t with $X_i^{(t)}$ sampled from distribution P_i.

In Figure 3 we show the probability of error versus the number of samples per item for the test score algorithm and the SAA method. We observe that for a fixed number of samples, the SAA method results in a larger error probability. Intuitively, this happens because the SAA method requires us to bound the error probability for *every* set S that contains at least one safe item whereas for the replication test score approach, it suffices to compare pairs of items i,j where $i \in A$ and $j \in B$. For the example under consideration, the number of samples per item needed to guarantee a given probability of error can be analytically characterized; we show this in Appendix H. We remark that even if the SAA method is employed in combination with an approximation algorithm such as greedy, it would still result in a higher error probability since these algorithms require more oracle queries. Consequently, it may be necessary to bound the error probability for a larger number of sets S.

In summary, our numerical results demonstrate the efficiency of the test score algorithm for different values of parameters and in comparison with the sample averaging approach.

5.2. Mean and Quantile Test Scores

As we already mentioned, the mean test scores are defined as expected values $a_i = \mathbf{E}[X_i]$. The quantile test scores are defined as $a_i = \mathbf{E}[X_i \mid P_i(X_i) \geq \theta]$, for a parameter $\theta \in [0,1]$. For the value of parameter $\theta = 0$, the quantile test score corresponds to mean test score. The quantile test score is defined as the conditional expectation of individual performance of an item conditional on it being larger than a quantile value. Note that quantile test scores, as defined here, are not quantile values following standard definition in statistics.

Neither mean test scores nor quantile test scores can guarantee a constant-factor approximation for the submodular function maximization problem. We demonstrate this by two simple examples that convey intuitive explanations on why these test scores can fail to provide desired guarantee. We then present tight approximation bounds for the CES utility functions.

Example 1 (mean test scores): Suppose that the utility is according to the best-shot function and that the selection is greedy using mean test scores. Suppose that there are two types of items: (a) deterministic performance items whose each individual performance is of value 1 with probability 1 and (b) random performance items whose individual performances are independent with expected value strictly smaller than 1 and a strictly positive probability of being larger than 1. Then, the algorithm will select all items to be those with deterministic performance. This is clearly suboptimal under the best-shot production where having selected an item with deterministic performance, the only way to increase the performance of a set of items with a non-zero probability is to select an item with random performance. Such an instance can be chosen such that the algorithm yields the utility that is only factor $O(1/k)$ of the optimum value.

Example 2 (quantile test scores): Suppose that the utility function is the sum of individual performances and consider greedy selection with respect to quantile test scores with threshold parameters $\theta_i = 1 - 1/k$. Suppose there are two types of items: (a) deterministic performance items whose individual performances are of value 1 with probability 1 and (b) random performance items whose individual performances are independent of value $a > 1$ with probability $p > 1/k$ and otherwise equal to zero. For random performance items, the mean test score is of value ap and the quantile test score is of value a. The algorithm will choose all items to be random performance items, which yields the utility of value kap. On the other hand, choosing items that output deterministic performance, yields the utility of value k. Since a and p can be chosen to be arbitrarily near to values 1 and $1/k$, respectively, we observe that the algorithm yields the utility that is $O(1/k)$ of the optimum value.

We next present a tight approximation bound for the CES utility function with parameter $r \geq 1$. Recall that the CES utility production provides an interpolation between two extreme cases: a linear function (for $r = 1$) and the best-shot function (as $r \to \infty$). Intuitively, we would expect that mean test scores would perform well for small values of parameter r, but that the approximation would get worse by increasing parameter r. The following result makes this intuition precise.

Proposition 11 **(mean test scores)**. *Suppose that the utility function u is according to the CES production function with parameter $r \geq 1$. For given cardinality parameter $k \geq 1$, let M be a set of k items in N with highest mean test scores. Then, we have*

$$u(M) \geq \left(\frac{1}{k}\right)^{1-\frac{1}{r}} u(\mathrm{OPT}).$$

Moreover, this bound is tight.

The proof of Proposition 11 is provided in Appendix I. The approximation factor decreases with the value of parameter r from value 1 for $r=1$ to value $1/k$ as r goes to infinity. The limit value $1/k$ conforms to the approximation factor obtained for the best-shot function in Kleinberg and Raghu (2018).

Intuitively, we would expect that quantile test scores would yield a good approximation guarantee for the CES utility function with large enough parameter r. This is because the quantile test scores guarantee a constant-factor approximation for the best-shot utility function (Kleinberg and Raghu 2018). The following result makes this intuition precise.

PROPOSITION 12 **(quantile test scores)**. *Suppose that the utility is according to the CES production function with parameter r and that the selection is greedy using quantile test scores with $\theta = 1 - c/k$ and $c > 0$. Then, we have*

(a) if $r = o(\log(k))$ and $r > 1$, the quantile test scores cannot guarantee a constant-factor approximation for any value of parameter $c > 0$;

(b) if $r = \Omega(\log(k))$, the quantile test scores with $c = 1$ guarantee a constant-factor approximation.

The proof of Proposition 12 is provided in Appendix J. The proposition establishes that quantile test scores can guarantee a constant-factor approximation if, and only if, the parameter r is larger than a threshold of value $\Theta(\log(k))$.

6. Conclusion

In this paper we presented a new algorithmic approach for the problem of stochastic submodular maximization by using test score algorithms. These algorithms are particularly appealing due to their simplicity and natural interpretation as their decisions are contingent only on individual item scores that are computed based on the distribution that captures the uncertainty in the respective item's performance. Although test score based methods have been studied in previous literature (Kleinberg and Raghu 2018), our work presents a new systematic framework for solving a broad class of stochastic combinatorial optimization problems by approximating complex set functions using simpler test score based sketch functions. By leveraging this framework, we show that it is possible to obtain good approximations under a natural extended diminishing returns condition, namely: (a) a constant factor approximation for the problem of maximizing a stochastic submodular function subject to a cardinality constraint, and (b) a logarithmic-approximation guarantee for the more general stochastic submodular welfare maximization problem. It is worth noting that since test score algorithms represent a more restrictive computational model, the guarantees obtained in this paper are not as good as those of the best known algorithms for both these problems. However, test score based approaches provide three key advantages over more traditional algorithms that make them highly desirable in practical situations relating to online platforms:

- *Scalability*: The test score of an item depends only on its own performance distribution. Therefore, when new items are added or existing items are removed from the ground set, this does not alter the scores of any other items. Since our algorithm selects items with the highest test scores, its output would only require simple swaps when the ground set changes.
- *Distributed Implementation*: Test score algorithms can be easily parallelized as the test score of an item can be computed independently of the performance distribution of other items.
- *Fewer Oracle Calls*: Test score algorithms only query the value of the function once per item—n oracle calls in total—which is an order of magnitude smaller than the number required by traditional approaches. Moreover, these oracle calls are simple in that they do not require drawing samples from the distributions of multiple items, which may be expensive.

Due to the prevalence of numeric quality scores for online recommendations (e.g., Agarwal et al. (2008), Adomavicius and Tuzhilin (2005)) and in team selection (Kleinberg and Raghu 2018), we used these two domains as running examples to illustrate the practical applications of test score algorithms. Although we focused specifically on these examples, there are other applications involving stochastic submodular functions that satisfy extended diminishing returns where test score algorithms are naturally appealing. As Koç and Morton (2014) point out, in resource constrained selection problems, practitioners prefer assigning priority scores to items and selecting the top items instead of employing an optimal algorithm. For example, our framework in (5) could be used to model a facility location problem with the concave objective function $\mathbf{E}[\sum_{j\in M} f_j(\sum_{i\in S} X_{i,j})]$ as in (Ahmed and Atamtürk 2011). Here, M is the set of markets and $X_{i,j}$ is a random demand that quantifies market j's preference for location i. Alternatively, in online advertising, a firm may seek to select a portfolio of channels or keywords to maximize the *reach* of its ad campaign, i.e., maximize $\mathbf{E}[1-\prod_{i\in S}(1-X_i)]$, where X_i is the indicator that channel i reaches a random targeted user (Danaher et al. 2010, Paulson et al. 2018). In these scenarios, test scores measure the importance of each location or channel and could be estimated via market surveys.

One possible direction for future work is to gain a better understanding of the limits and capabilities of test scores by deriving lower bounds on the performance of such algorithms. In this regard, an important open question is whether the logarithmic-approximation guarantee for the welfare maximization problem presented in this paper is tight for the class of test score based algorithms. In Appendix K, we present some negative examples which suggest that it may not be possible to obtain a constant-factor approximation for the welfare maximization problem using test score algorithms based on function sketching. Similarly, it would be interesting to design test score algorithms for set functions that do not satisfy the extended diminishing returns condition introduced in this work. Finally, it would also be of interest to study approximation guarantees when using statistical estimators for test scores, and not expected values as in this paper.

References

Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering* (6):734–749.

Agarwal D, Chen BC, Elango P, Motgi N, Park ST, Ramakrishnan R, Roy S, Zachariah J (2008) Online models for content optimization. *Proceedings of the 21st International Conference on Neural Information Processing Systems*, 17–24, NIPS'08.

Agrawal S, Zadimoghaddam M, Mirrokni V (2018) Proportional allocation: Simple, distributed, and diverse matching with high entropy. *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 99–108 (PMLR).

Ahmed S, Atamtürk A (2011) Maximizing a class of submodular utility functions. *Mathematical Programming* 128(1):149–169.

Asadpour A, Nazerzadeh H (2016) Maximizing stochastic monotone submodular functions. *Management Science* 62(8):2374–2391.

Asadpour A, Nazerzadeh H, Saberi A (2008) Stochastic submodular maximization. *International Workshop on Internet and Network Economics (WINE)*, 477–489.

Badanidiyuru A, Vondrák J (2014) Fast algorithms for maximizing submodular functions. *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1497–1514, SODA '14.

Balcan M, Harvey NJA (2011) Learning submodular functions. *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, 793–802.

Balkanski E, Rubinstein A, Singer Y (2019) An exponential speedup in parallel running time for submodular maximization without loss in approximation. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 283–302, SODA '19.

Besbes O, Gur Y, Zeevi A (2015) Optimization in online content recommendation services: Beyond click-through rates. *Manufacturing & Service Operations Management* 18(1):15–33.

Bhowmik A, Borkar V, Garg D, Pallan M (2014) Submodularity in team formation problem. *Proceedings of the 2014 SIAM International Conference on Data Mining*, 893–901.

Calinescu G, Chekuri C, Pál M, Vondrák J (2011) Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing* 40(6):1740–1766.

Chien CF (2002) A portfolio–evaluation framework for selecting r&d projects. *R&D Management* 32(4):359–368.

Cohavi K, Dobzinski S (2017) Faster and simpler sketches of valuation functions. *ACM Trans. Algorithms* 13(3):30:1–30:9.

Cohen MC, Keller PW, Mirrokni V, Zadimoghaddam M (2019) Overcommitment in cloud services: Bin packing with chance constraints. *Management Science* 65(7):3255–3271.

Danaher PJ, Lee J, Kerbache L (2010) Optimal internet media selection. *Marketing Science* 29(2):336–347.

Dixit AK, Stiglitz JE (1977) Monopolistic Competition and Optimum Product Diversity. *American Economic Review* 67(3):297–308.

Ene A, Nguyen HL (2019) Submodular maximization with nearly-optimal approximation and adaptivity in nearly-linear time. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 274–282, SODA '19.

Feige U (1998) A threshold of ln n for approximating set cover. *J. ACM* 45(4):634–652.

Feldman M, Zenklusen R (2018) The submodular secretary problem goes linear. *SIAM Journal on Computing* 47(2):330–366.

Feldman V, Vondrák J (2014) Optimal bounds on approximation of submodular and XOS functions by juntas. *Information Theory and Applications Workshop, ITA 2014, San Diego, CA, USA, February 9-14, 2014*, 1–10.

Fu R, Subramanian A, Venkateswaran A (2016) Project characteristics, incentives, and team production. *Management Science* 62(3):785–801.

Goel A, Guha S, Munagala K (2006) Asking the right questions: model-driven optimization using probes. *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Conference*, 203–212.

Goemans MX, Harvey NJA, Iwata S, Mirrokni V (2009) Approximating submodular functions everywhere. *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 535–544, SODA '09.

Gotovos A, Hassani SH, Krause A (2015) Sampling from probabilistic submodular models. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, 1945–1953, NIPS'15.

Hassidim A, Singer Y (2017) Submodular optimization under noise. Kale S, Shamir O, eds., *Proceedings of COLT 2017*, volume 65 of *Proceedings of Machine Learning Research*, 1069–1122.

Henriksen AD, Traynor AJ (1999) A practical R&D project-selection scoring tool. *IEEE Transactions on Engineering Management* 46(2):158–170.

Herbrich R, Minka T, Graepel T (2006) Trueskill$^{\text{TM}}$: A bayesian skill rating system. *Proceedings of the 19th International Conference on Neural Information Processing Systems*, 569–576, NIPS'06.

Hoeffding W (1963) Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* 58(301):13–30.

Iyer R, Bilmes J (2013) Submodular optimization with submodular cover and submodular knapsack constraints. *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, 2436–2444, NIPS'13.

Kempe D, Kleinberg JM, Tardos É (2015) Maximizing the spread of influence through a social network. *Theory of Computing* 11:105–147.

Kleinberg J, Raghu M (2018) Team performance with test scores. *ACM Transactions on Economics and Computation (TEAC)* 6(3-4):17.

Kleinberg JM, Oren S (2011) Mechanisms for (mis)allocating scientific credit. *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, 529–538.

Kleywegt A, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12(2):479–502.

Koç A, Morton DP (2014) Prioritization via stochastic optimization. *Management Science* 61(3):586–603.

Korula N, Mirrokni V, Zadimoghaddam M (2018) Online submodular welfare maximization: Greedy beats 1/2 in random order. *SIAM Journal on Computing* 47(3):1056–1086.

Krause A, Golovin D (2014) *Submodular Function Maximization*, 71–104 (Cambridge University Press).

Lehmann B, Lehmann D, Nisan N (2006) Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior* 55(2):270–296.

Li H (2011) *Learning to Rank for Information Retrieval and Natural Language Processing* (Morgan & Claypool).

Manning C, Raghavan P, Schütze H (2010) Introduction to information retrieval. *Natural Language Engineering* 16(1):100–103.

Mehta A, et al. (2013) Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science* 8(4):265–368.

Nemhauser G, Wolsey L, Fisher M (1978) An analysis of approximations for maximizing submodular set functions—i. *Math. Programming* 14(1):265–294.

Paulson C, Luo L, James GM (2018) Efficient large-scale internet media selection optimization for online display advertising. *Journal of Marketing Research* 55(4):489–506.

Shapiro A, Nemirovski A (2005) *On Complexity of Stochastic Programming Problems*, 111–146 (Boston, MA: Springer US).

Singla A, Tschiatschek S, Krause A (2016) Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2037–2043, AAAI'16.

Sviridenko M, Vondrák J, Ward J (2017) Optimal approximation for submodular and supermodular optimization with bounded curvature. *Mathematics of Operations Research* 42(4):1197–1218.

Swamy C, Shmoys DB (2012) Sampling-based approximation algorithms for multistage stochastic optimization. *SIAM Journal on Computing* 41(4):975–1004.

Vondrak J (2008) Optimal approximation for the submodular welfare problem in the value oracle model. *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, 67–74, STOC '08.

Yuan R, Zhao L, Wang W (2007) Cooperation and competition dynamics in an online game community. *International Conference on Online Communities and Social Computing*, 475–484 (Springer).

Appendix. Proofs and Additional Results

A. Validation of the Extended Diminishing Returns Property

It is easy to verify that all the value functions defined in Section 2.3 are symmetric, monotone submodular value functions. We next show that they also satisfy the extended diminishing returns condition, defined in Definition 1. Formally, we need to check that a value function g is such that for any given $v \in \mathbf{R}_+$ with a non-empty preimage under g, there exists $\mathbf{z} \in \mathbf{R}_+^{n-1}$ such that $g(\mathbf{z}) = v$ and for all $\mathbf{y} \in \mathbf{R}_+^{n-1}$ with $g(\mathbf{y}) \leq g(\mathbf{z})$, the following condition holds:

$$g(z_1,\dots,z_{n-1},x) - g(z_1,\dots,z_{n-1}) \leq g(y_1,\dots,y_{n-1},x) - g(y_1,\dots,y_{n-1}), \text{ for all } x \in \mathbf{R}_+. \tag{13}$$

We first prove that all of the functions defined in Section 2.3 except the top-r function with $r > 1$ satisfy a stronger version of the above condition. More specifically, we show that for any $v \in \mathbf{R}_+$ and $\mathbf{z}$ such $g(\mathbf{z}) = v$, and $\mathbf{y}$ satisfying $g(\mathbf{y}) \leq g(\mathbf{z})$, the following is true:

$$g(z_1,\dots,z_{n-1},z) - g(z_1,\dots,z_{n-1}) \leq g(y_1,\dots,y_d,x) - g(y_1,\dots,y_d), \text{ for all } x \in \mathbf{R}_+. \tag{14}$$

Note for any given vector $\mathbf{z} \in \mathbf{R}_+^d$ with $d < n$, we write $g(\mathbf{z})$ in lieu of $g(\mathbf{z},0,0,\dots,0)$. We begin by proving that all of the functions defined in Section 2.3 except top-r satisfy the stronger condition as per (14).

Total production: $g(\mathbf{x}) = \bar{g}(\sum_{i=1}^n x_i)$. In this case, $g(\mathbf{y}) \leq g(\mathbf{z})$ implies that $\sum_{i=1}^{n-1} y_i \leq \sum_{i=1}^{n-1} z_i$ by monotonicity and thus, (14) is equivalent to

$$\bar{g}\left(\sum_{i=1}^{n-1} y_i + x\right) - \bar{g}\left(\sum_{i=1}^{n-1} y_i\right) \geq \bar{g}\left(\sum_{i=1}^{n-1} z_i + x\right) - \bar{g}\left(\sum_{i=1}^{n-1} z_i\right), \text{ for all } x \in \mathbf{R}_+.$$

Let $y = \sum_{i=1}^{n-1} y_i$ and $z = \sum_{i=1}^{n-1} z_i$. With this new notation, the extended diminishing returns condition is equivalent to saying that for all $y, z \in \mathbf{R}_+$ such that $y \leq z$,

$$\bar{g}(y+x) - \bar{g}(y) \geq \bar{g}(z+x) - \bar{g}(z), \text{ for all } x \in \mathbf{R}_+$$

which obviously holds true because $\bar{g}$ is assumed to be a monotone increasing and concave function.

Best-shot: $g(\mathbf{x}) = \max\{x_1, x_2, \dots, x_n\}$. In this case, $g(\mathbf{y}) \leq g(\mathbf{z})$ implies that

$$\max\{y_1,\dots,y_{n-1}\} \leq \max\{z_1,\dots,z_{n-1}\}$$

and (14) is equivalent to

$$\max\{y_1,\dots,y_{n-1},x\} - \max\{y_1,\dots,y_{n-1}\} \geq \max\{z_1,\dots,z_{n-1},x\} - \max\{z_1,\dots,z_{n-1}\} \text{ for all } x \in \mathbf{R}_+.$$

We consider three different cases:

- Case 1: $x \geq \max\{z_1,\dots,z_{n-1}\} \geq \max\{y_1,\dots,y_{n-1}\}$. In this case, $\max\{\mathbf{y},x\} - \max\{\mathbf{y}\} = x - \max\{\mathbf{y}\} \geq x - \max\{\mathbf{z}\} = \max\{\mathbf{z},x\} - \max\{\mathbf{z}\}$. Hence, the extended diminishing returns condition holds.
- Case 2: $\max\{y_1,\dots,y_{n-1}\} \leq z < \max\{z_1,\dots,z_{n-1}\}$. In this case, condition (14) is equivalent to $x \geq \max\{y_1,\dots,y_{n-1}\}$, which holds by assumption.
- Case 3: $z < \max\{y_1,\dots,y_{n-1}\}$. In this case, condition (14) is equivalent to $0 \geq 0$ and thus trivially holds.

CES: $g(\mathbf{x}) = (\sum_{i=1}^n x_i^r)^{1/r}$*, for parameter* $r \geq 1$*.* Let $y = \sum_{i=1}^{n-1} y_i^r$, $z = \sum_{i=1}^r z_i^r$ and $w = x^r$. Condition (14) is equivalent to

$$(y+w)^{1/r} - y^{1/r} \geq (z+w)^{1/r} - z^{1/r}$$

while $g(\mathbf{y}) \leq g(\mathbf{z})$ is equivalent to $y \leq z$. Since $r \geq 1$, the function $f(y) = y^{1/r}$ is an increasing concave function. Hence, it follows that condition (14) holds as long as $g(\mathbf{y}) \leq g(\mathbf{z})$.

Success-probability: $g(\mathbf{x}) = 1 - \prod_{i=1}^n (1 - p(x_i))$*.* By simple algebra, condition (14) is equivalent to

$$\prod_{i=1}^{n-1} p(y_i)(1-p(x)) \geq \prod_{i=1}^{n-1} p(z_i)(1-p(x))$$

while $g(\mathbf{y}) \leq g(\mathbf{z})$ is equivalent to

$$\prod_{i=1}^{n-1} p(y_i) \geq \prod_{i=1}^{n-1} p(z_i).$$

Hence, condition (14) holds as long as $g(\mathbf{y}) \leq g(\mathbf{z})$.

Finally, we prove that the top-r function satisfies (13) for $r > 1$. Recall that the top-1 function coincides with the best-shot function, for which we already established that the extended diminishing returns condition holds.

Top-r: $g(\mathbf{x}) = \sum_{i=1}^r x_{(i)}$*, where* $x_{(i)}$ *is the i–th largest element in* $\mathbf{x}$*.* Fix $v \in \mathbf{R}_+$. Without loss of generality, suppose that $n-1 \geq r$ and define $\mathbf{z} = (z_1, \ldots, z_{n-1}) \in \mathbf{R}^{n-1}$ such that $z_j = v/r$ for $1 \leq j \leq r$ and $z_j = 0$ for all $r < j \leq n-1$.[5] Clearly, $g(\mathbf{z}) = v$.

Let $\mathbf{y} \in \mathbf{R}_+^{n-1}$ be any point such that $g(\mathbf{y}) \leq g(\mathbf{z})$. We prove (13) for the following two different cases:

- Case 1: $x \geq v/r$: In this case, $g(\mathbf{z}, x) - g(\mathbf{z}) = x - v/r$. Since $g(\mathbf{y}) \leq g(\mathbf{z})$, it must be that the r-th largest element in $\mathbf{y}$, i.e. $y_{(r)}$, is smaller than or equal to $g(\mathbf{z})/r = v/r$. Thus, we have that $g(\mathbf{y}, x) - g(\mathbf{y}) = x - y_{(r)} \geq x - g(\mathbf{z})/r = g(\mathbf{z}, x) - g(\mathbf{z})$ and so, the claim follows.
- Case 2: $x \leq v/r$: The claim trivially follows in this case because $g(\mathbf{z}, x) = g(\mathbf{z})$ and so, $g(\mathbf{z}, x) - g(\mathbf{z}) = 0$, whereas $g(\mathbf{y}, x) - g(\mathbf{y}) \geq 0$.

B. Proof of Lemma 3

We first note the following inequalities

$$u(\mathrm{OPT}) \leq u(S^*) + u(\mathrm{OPT} \setminus S^*) \leq u(S^*) + q\bar{v}(\mathrm{OPT} \setminus S^*).$$

The first inequality is by the fact that all submodular functions are subadditive, i.e. for any submodular set function u, it holds $u(A \cup B) \leq u(A) + u(B)$. The second inequality is by the sketch upper bound.

Now, consider any set T of cardinality k such that $\mathrm{OPT} \setminus S^* \subseteq T$ that is disjoint from S^*, i.e. $S^* \cap T = \emptyset$. By the condition of the lemma, we have that $\bar{v}(T) \leq \underline{v}(S^*)$ and $p\underline{v}(S^*) \leq u(S^*)$. Therefore, we have

$$u(\mathrm{OPT}) \leq u(S^*) + q\bar{v}(T) \leq u(S^*) + q\underline{v}(S^*) \leq u(S^*) + \frac{q}{p} u(S^*)$$

which completes the proof.

[5] The proof when $r > n-1$ is trivial because $g(\mathbf{y}, x) - g(\mathbf{y}) = g(\mathbf{z}, x) - g(\mathbf{z}) = x$.

C. Proof of Lemma 5

Suppose that a set function u has (p,q)-good test scores $a_1, a_2, \dots, a_n$, i.e. for every $S \subseteq N$ such that $|S| = k$,

$$p \min\{a_i \mid i \in S\} \le u(S) \le q \max\{a_i \mid i \in S\}. \tag{15}$$

Let $r_1, r_2, \dots, r_n$ be the replication test scores, i.e.[6]

$$r_i = \mathbf{E}[g(X_i^{(1)}, \dots, X_i^{(k)}, \phi, \dots, \phi)] = u(\{i^{(1)}, \dots, i^{(k)}\}) \tag{16}$$

where $X_i^{(1)}, X_i^{(2)}, \dots, X_i^{(k)}$ are independent random variables with distribution P_i and $i^{(1)}, i^{(2)}, \dots, i^{(k)}$ are independent replicas of item i.

By assumption, $a_1, a_2, \dots, a_n$ are (p,q)-good test scores, hence

$$pa_i \le u(\{i^{(1)}, \dots, i^{(k)}\}) \le qa_i. \tag{17}$$

From (15), (16), and (17), for every $S \subseteq N$ such that $|S| = k$,

$$\frac{p}{q} \min\{r_i \mid i \in S\} \le p \min\{a_i \mid i \in S\} \le u(S) \le q \max\{a_i \mid i \in S\} \le \frac{q}{p} \max\{r_i \mid i \in S\}$$

which implies that the replication test scores are $(p/q, q/p)$-good test scores.

D. Proof of Lemma 6

We first prove the lower bound and then the upper bound as follows.

Proof of the lower bound. Without loss of generality, let us consider the set $S = \{1, 2, \dots, k\}$ and assume that $a_1 = \min\{a_i \mid i \in S\}$. We claim that

$$u(\{1, \dots, j\}) \ge \left(1 - \frac{1}{k}\right) u(\{1, \dots, j-1\}) + \frac{1}{k} a_1 \text{ for all } j \in \{1, 2, \dots, k\}. \tag{18}$$

We prove this claim sequentially beginning with $j = 1$ up to $j = k$. From this, we can use a cascading argument to show that $u(S) \ge (1 - (1 - \frac{1}{k})^k) a_1 \ge (1 - \frac{1}{e}) a_1$.

We next prove the claim in (18). For $j = 1$, since u is a non-negative, monotone submodular set function, we have

$$u(\{1\}) = \frac{1}{k} \sum_{t=1}^{k} u(\{1^{(t)}\}) \ge \frac{1}{k} u(\{1^{(1)}, \dots, 1^{(k)}\}) = \frac{1}{k} a_1. \tag{19}$$

For $j > 1$, we have

$$\begin{aligned}
u(\{1, \dots, j\}) &= u(\{1, \dots, j-1\}) + [u(\{1, \dots, j\}) - u(\{1, \dots, j-1\})] \\
&\overset{(a)}{\ge} u(\{1, \dots, j-1\}) + \frac{1}{k} [u(\{1, \dots, j-1, j^{(1)}, \dots, j^{(k)}\}) - u(\{1, \dots, j-1\})] \\
&\overset{(b)}{\ge} u(\{1, \dots, j-1\}) + \frac{1}{k} [u(\{j^{(1)}, \dots, j^{(k)}\}) - u(\{1, \dots, j-1\})] \\
&= \left(1 - \frac{1}{k}\right) u(\{1, \dots, j-1\}) + \frac{1}{k} a_j \\
&\ge \left(1 - \frac{1}{k}\right) u(\{1, \dots, j-1\}) + \frac{1}{k} a_1
\end{aligned} \tag{20}$$

[6] Hereinafter, we slightly abuse the notation by writing $u(S)$ for a set of item i replicas $S = \{i^{(1)}, \dots, i^{(k)}\}$ while u is defined as a set function over 2^N. A proper definition would extend the definition of u over $2^{\tilde{N}}$ where $\tilde{N}$ includes n instances of each item $i \in N$ but this would be at the expense of more complex notation.

where (a) follows by submodularity of u and (b) follows by non-negativity and monotonicity of u. This completes the proof of (18). We now proceed with the cascading argument:

$$\begin{aligned}
u(\{1,\ldots,k\}) &\geq \left(1-\frac{1}{k}\right)u(\{1,\ldots,k-1\})+\frac{1}{k}a_1\\
&\geq \left(1-\frac{1}{k}\right)^2 u(\{1,\ldots,k-2\})+\left(1-\frac{1}{k}\right)\frac{a_1}{k}+\frac{a_1}{k}\\
&\geq \cdots\\
&\geq \frac{a_1}{k}\left(\sum_{j=0}^{k-1}\left(1-\frac{1}{k}\right)^j\right)\\
&\geq a_1\left(1-\left(1-\frac{1}{k}\right)^k\right)\\
&\geq \left(1-\frac{1}{e}\right)a_1.
\end{aligned}$$

For the last step, we use the fact that $(1-1/k)^k \leq 1/e$, for all $k \geq 1$.

Proof of the upper bound. Again, without loss of generality, assume that $S=\{1,2,\ldots,k\}$ and $a_1 \leq a_2 \leq \cdots \leq a_k$. Recall that the value function g is defined on $\mathbf{R}^n$. Recall that we slightly abuse the notation by writing $g(\mathbf{y})$ to denote $g(\mathbf{y},\phi,\ldots,\phi)$, for any vector $\mathbf{y}$ of dimension $1 \leq d < n$, where ϕ is some minimal-value element defined in Section 2. Moreover, for convenience, we will assume that the value function g is continuous on any given dimension.

Define g_i^{max} to be the maximum value of the submodular function g on a vector of dimension i, i.e.,

$$g_i^{\max} = \max_{z_1,z_2,\ldots,z_i \in \mathbf{R}_+} g(z_1,z_2,\ldots,z_i).$$

Suppose that $v=\min\{ca_k, g_{k-1}^{\max}\}$, for some constant $c>1$ whose value we will determine later. We first claim that there exists a vector $\mathbf{z}$ such that $g(\mathbf{z})=v$. Our proof will leverage this vector $\mathbf{z}$ as follows. We consider a fictitious set of items S^* whose individual performances correspond to $\mathbf{z}$ and show that the marginal benefit of adding an item $i \in N$ to this fictitious set is at most twice the marginal value of adding item i to a set comprising of $k-1$ replicas of item i. This allows us to establish an upper bound in terms of the test scores. Although $g(\mathbf{z})=v=ca_k$ is sufficient for our proof to hold, it is possible that the function g is capped at a value smaller than ca_k and there does not exist any $\mathbf{z}$ satisfying $g(\mathbf{z})=ca_k$. To handle this corner case, we define v to be the minimum of ca_k and $g_{k-1}^{\max}$.

We now prove the above claim that v has a non-empty preimage under g. When $v=g_{k-1}^{\max}$, the claim follows trivially since by the definition of $g_i^{\max}$, there exists a $(k-1)$-dimension vector for which the function value is $g_{k-1}^{\max}$. On the other hand, when $ca_k < g_{k-1}^{\max}$, this comes from continuity arguments since we know that there exist points in $\mathbf{R}_+^{k-1}$ where g evaluates to values greater than and smaller than v respectively. In summary, there exists at least one point where the function evaluates to v. Since g satisfies the extended diminishing returns condition, we can abuse notation and infer from the definition that there exists a vector[7] $z \in \mathbf{R}_+^{n-1}$ such that $g(\mathbf{z})=v$ and for any $\mathbf{y} \in \mathbf{R}_+^{k-1}$ having $g(\mathbf{y}) \leq g(\mathbf{z})$, it must be that

$$g(\mathbf{z},x)-g(\mathbf{z}) \leq g(\mathbf{y},x)-g(\mathbf{y}), \text{ for all } x \in \mathbf{R}_+. \tag{21}$$

[7] Note that some elements of this vector can be ϕ or zero.

It is worth pointing out that while Definition 1 guarantees that (21) holds when the vector $\mathbf{y}$ is of dimension $n-1$, one can simply start with a $(k-1)$-dimension vector $\mathbf{y}$ and simply pad a sufficient number of ϕ elements to arrive upon a $(n-1)$-dimension vector whose value is still $g(\mathbf{y})$. Therefore, let $\mathbf{z}=(z_1,z_2,\dots,z_{n-1})^\top$ be an arbitrary vector such that $g(\mathbf{z})=v$ and that it satisfies (21) for any $\mathbf{y}\in\mathbf{R}_+^{k-1}$, $x\geq 0$ as long as $g(\mathbf{y})\leq g(\mathbf{z})$. Let $S^*=\{q_1,q_2,\dots,q_{n-1}\}$ be a set of (fictitious) items such that $X_{q_j}=z_j$ with probability 1 (performance of each of these fictitious items is deterministic). Therefore, the performance of the set of items S^* is given by

$$u(S^*)=g(\mathbf{z})=\min\{ca_k,g_{k-1}^{max}\}.$$

Since u is a non-negative, increasing and submodular function, we have

$$\begin{aligned}
u(S) &\leq u(S^*\cup S) &(22)\\
&\leq u(S^*)+\sum_{i=1}^{k}\left(u(S^*\cup\{i\})-u(S^*)\right) &(23)\\
&\leq ca_k+\sum_{i=1}^{k}\left(u(S^*\cup\{i\})-u(S^*)\right). &(24)
\end{aligned}$$

Let $X_i^{(1)},X_i^{(2)},\dots,X_i^{(k)}$ be independent random variables with distribution P_i. Let $X_i=X_i^{(k)}$ and $\mathbf{Y}_i=(X_i^{(1)},X_i^{(2)},\dots,X_i^{(k-1)})^\top$. Note that

$$\begin{aligned}
u(S^*\cup\{i\})-u(S^*) &= \mathbf{E}\left[g(\mathbf{z},X_i)-g(\mathbf{z})\right] &(25)\\
&\overset{(a)}{=} \mathbf{E}\left[g(\mathbf{z},X_i)-g(\mathbf{z})\mid g(\mathbf{Y}_i)\leq g(\mathbf{z})\right] &(26)\\
&\overset{(b)}{\leq} \mathbf{E}\left[g(\mathbf{Y}_i,X_i)-g(\mathbf{Y}_i)\mid g(\mathbf{Y}_i)\leq g(\mathbf{z})\right] &(27)\\
&\leq \frac{u\left(\{i^{(1)},\dots,i^{(k)}\}\right)-u\left(\{i^{(1)},\dots,i^{(k-1)}\}\right)}{\mathbf{Pr}\left[g(\mathbf{Y}_i)\leq g(\mathbf{z})\right]} &(28)\\
&\overset{(c)}{\leq} \frac{1}{\mathbf{Pr}\left[g(\mathbf{Y}_i)\leq g(\mathbf{z})\right]}\frac{a_i}{k} &(29)\\
&\overset{(d)}{\leq} \left(1-\frac{1}{c}\right)^{-1}\frac{a_k}{k}, &(30)
\end{aligned}$$

where (a) comes from the fact that, by definition, X_i and $\mathbf{Y}_i$ are independent; the inequality (b) follows from the extended diminishing returns condition defined in (21) for $\mathbf{y}=\mathbf{Y}_i$–note that for any instantiation $\mathbf{Y}_i$ where $g(\mathbf{Y}_i)\leq g(\mathbf{z})$, extended diminishing returns tells us that $g(\mathbf{z},X_i)-g(\mathbf{z})\leq g(\mathbf{Y}_i,X_i)-g(\mathbf{Y}_i)$ for all X_i, thus taking the expectation over all $\mathbf{Y}_i,X_i$ conditional upon $g(\mathbf{Y}_i)\leq g(\mathbf{z})$ gives us (b); inequality (c) can be shown using only the definition of submodularity as can be seen via the below sequence of inequalities:

$$\begin{aligned}
u\left(\{i^{(1)},\dots,i^{(k)}\}\right)-u\left(\{i^{(1)},\dots,i^{(k-1)}\}\right) &\leq \frac{1}{k}\sum_{j=0}^{k-1}\left(u(\{i^{(1)},\dots,i^{(j)},i^{(k)}\})-u(\{i^{(1)},\dots,i^{(j)}\})\right)\\
&=\frac{1}{k}\sum_{j=0}^{k-1}\left(u(\{i^{(1)},\dots,i^{(j)},i^{(j+1)}\})-u(\{i^{(1)},\dots,i^{(j)}\})\right)\\
&=\frac{1}{k}u(\{i^{(1)},\dots,i^{(k)}\})\\
&=\frac{a_i}{k}.
\end{aligned}$$

It remains to prove (d) in (30), which follows by the fact $a_i \geq a_k$ for all $i \in \{1, 2 \ldots, k\}$ and showing that $\mathbf{Pr}\big[g(\mathbf{Y}_i) \leq g(\mathbf{z})\big] \geq 1 - 1/c$. Recall that $g(\mathbf{z}) = \min\{ca_k, g_{k-1}^{max}\}$. Let us proceed by separately considering two cases depending on the value of $g(\mathbf{z})$. If $g(\mathbf{z}) = g_{k-1}^{max}$, then $\mathbf{Pr}\big[g(\mathbf{Y}_i) \leq g(\mathbf{z})\big] = 1$ trivially. This is because by definition g_{k-1}^{max} is the maximum value that the function can take for any vector of dimension $k-1$. On the other hand, when $g(\mathbf{z}) = ca_k$, we can apply Markov's inequality to obtain

$$\mathbf{Pr}[g(\mathbf{Y}_i) \geq ca_k] \leq \frac{\mathbf{E}[g(\mathbf{Y}_i)]}{ca_k} \leq \frac{\mathbf{E}[g(\mathbf{Y}_i, X_i)]}{ca_k} \leq \frac{1}{c}.$$

Hence, it follows $\mathbf{Pr}[g(\mathbf{Y}_i) \leq ca_k] \geq 1 - \mathbf{Pr}[g(\mathbf{Y}_i) \geq ca_k] \geq 1 - 1/c$. Combining this with (24) and (30), we obtain $u(S) \leq ca_k + (1-1/c)^{-1} a_k = (c^2/(c-1)) a_k$. Since we can choose c arbitrarily, by taking $c = 2$, we obtain $u(S) \leq 4a_k$, which proves the upper bound.

E. Proof of Lemma 8

Suppose that S^* is the optimum solution to the submodular welfare maximization problem with the utility functions $v_1, v_2, \ldots, v_m$, and S is such that $v(S) \geq \alpha v(S^*)$. Then,

$$\begin{aligned}
u(\mathrm{OPT}) &= \sum_{j=1}^{m} u_j(\mathrm{OPT}_j) \\
&\leq q \sum_{j=1}^{m} v_j(\mathrm{OPT}_j) \\
&\leq q \sum_{j=1}^{m} v_j(S_j^*) \quad \text{(by definition of } S^*\text{)} \\
&\leq \frac{1}{\alpha} q \sum_{j=1}^{m} v_j(S_j) \\
&\leq \frac{1}{p\alpha} q \sum_{j=1}^{m} u_j(S_j).
\end{aligned}$$

F. Proof of Lemma 9

It suffices to consider an arbitrary partition j. To simplify the presentation, with a slight abuse of notation, we omit the index j in our notation.

Let $a_1^r, a_2^r, \ldots, a_n^r$ denote replication test scores for parameter r. For any set $S \subseteq N$ such that $|S| = k$, let $\pi(S) = (\pi_1(S), \pi_2(S), \ldots, \pi_k(S))$ be a permutation of the elements of S defined in (11).

Let v be a set function, which for any $S \subseteq N$ such that $|S| = k$ is defined by

$$v(S) = \sum_{r=1}^{k} \frac{1}{r} a_{\pi_r(S)}^r. \tag{31}$$

We need to establish the following relations, for every $S \subseteq N$,

$$u(S) \geq \frac{1}{2(\log(k)+1)} v(S) \tag{32}$$

and

$$u(S) \leq 6v(S). \tag{33}$$

Proof of lower bound (32) Suppose that S is of cardinality k and define

$$\tau := \arg\max_t a^t_{\pi_t(S)}.$$

We begin by noting the following basic property of replication test scores.

Lemma 13. *For replication test scores $a^r_1, a^r_2, \dots, a^r_n$ for $1 \le r \le k$, for every item $i \in \{1, 2, \dots, k\}$, the following relations hold:*

$$\frac{a^s_i}{s} \ge \frac{a^t_i}{t}, \text{ for } 1 \le s \le t \le k.$$

The assertion in Lemma 13 follows easily by the diminishing increments property of replication test scores a^r_i with respect to parameter r.

In our proof, we will also need the following lemma:

Lemma 14. *For every set $S \subseteq N$ such that $|S| = k$ and ordering of items of this set $\pi(S) = (\pi_1(S), \pi_2(S), \dots, \pi_k(S))$, the following relation holds:*

$$\frac{1}{\tau}\sum_{r=1}^{\tau} a^r_{\pi_r(S)} \ge \frac{1}{2} a^\tau_{\pi_\tau(S)}.$$

The proof of the lemma is as follows. For every $r \in \{1, 2, \dots, \tau\}$, we have

$$\frac{a^r_{\pi_r(S)}}{r} \ge \frac{a^r_{\pi_\tau(S)}}{r} \ge \frac{a^\tau_{\pi_\tau(S)}}{\tau}$$

where the first inequality is by definition of $\pi(S)$ and the second inequality is by Lemma 13. Hence, we have

$$\sum_{r=1}^{\tau} a^r_{\pi_r(S)} \ge \frac{a^\tau_{\pi_\tau(S)}}{\tau}\sum_{r=1}^{\tau} r \ge \frac{a^\tau_{\pi_\tau(S)}}{\tau}\frac{\tau(\tau+1)}{2} \ge a^\tau_{\pi_\tau(S)}\frac{\tau}{2}$$

which corresponds to the claim of the lemma.

Lemma 15. *For every $S \subseteq N$, the following holds:*

$$u(S) \ge \frac{1}{\tau}\sum_{r=1}^{\tau} a^r_{\pi_r(S)}.$$

The proof of Lemma 15 is by induction as we show next. The inductive statement is $u(\{\pi_1(S), \dots, \pi_r(S)\}) \ge \frac{1}{r}\sum_{s=1}^{r} a^s_{\pi_s(S)}$ for every $r \in \{1, 2, \dots, \tau\}$. Base case: $r = 1$. The base case indeed holds because by definition of replication test scores $u(\{\pi_1(S)\}) = a^1_{\pi_1(S)}$. Inductive step: assume that the statement is true up to $r - 1$ and we need to show that it holds for r. We have the following relations:

$$\begin{aligned}
&u(\{\pi_1(S), \dots, \pi_r(S)\}) - u(\{\pi_1(S), \dots, \pi_{r-1}(S)\}) \\
&= \frac{1}{r}\Big(u(\{\pi_1(S), \dots, \pi_{r-1}(S), \pi_r(S)^{(1)}\}) + \dots + u(\{\pi_1(S), \dots, \pi_{r-1}(S), \pi_r(S)^{(r)}\}) - ru(\{\pi_1(S), \dots, \pi_{r-1}(S)\})\Big) \\
&\ge \frac{1}{r}\Big(u(\{\pi_1(S), \dots, \pi_{r-1}(S), \pi_r(S)^{(1)}, \dots, \pi_r(S)^{(r)}\}) - u(\{\pi_1(S), \dots, \pi_{r-1}(S)\})\Big) \\
&\ge \frac{1}{r}\Big(u(\{\pi_r(S)^{(1)}, \dots, \pi_r(S)^{(r)}\}) - u(\{\pi_1(S), \dots, \pi_{r-1}(S)\})\Big) \\
&= \frac{a^r_{\pi_r(S)}}{r} - \frac{u(\{\pi_1(S), \dots, \pi_{r-1}(S)\})}{r}
\end{aligned}$$

where the first and second inequality is by submodularity and monotonicity of the set function u, respectively.

From the inductive hypothesis, we know that $u(\{\pi_1(S),\ldots,\pi_{r-1}(S)\}) \geq \frac{1}{r-1}\sum_{s=1}^{r-1} a^s_{\pi_s(S)}$, so we add $u(\{\pi_1(S),\ldots,\pi_{r-1}(S)\})$ to both sides of the above equation and obtain

$$u(\{\pi_1(S),\ldots,\pi_r(S)\}) \geq \frac{a^r_{\pi_r(S)}}{r} + \frac{r-1}{r}u(\{\pi_1(S),\ldots,\pi_{r-1}(S)\}) \geq \frac{1}{r}\sum_{s=1}^{r} a^s_{\pi_s(S)}$$

which proves the claim of Lemma 15.

Now, combining Lemma 14 and Lemma 15, we obtain $u(S) \geq a^\tau_{\pi_\tau(S)}/2$.

Finally, we conclude the lower bound as follows:

$$u(S) \geq \frac{1}{2}a^\tau_{\pi_\tau(S)} = \frac{a^\tau_{\pi_\tau(S)}}{2}\frac{1+\frac{1}{2}+\ldots+\frac{1}{k}}{1+\frac{1}{2}+\ldots+\frac{1}{k}} \geq \frac{a^1_{\pi_1(S)} + \frac{a^2_{\pi_2(S)}}{2} + \ldots + \frac{a^k_{\pi_k(S)}}{k}}{2(\log(k)+1)} = v(S)$$

where in the last inequality we use the facts that $a^\tau_{\pi_\tau(S)} \geq a^r_{\pi_r(S)}$ for all r, and $1+1/2+\cdots+1/k \leq \log(k)+1$, for all $k \geq 1$.

Proof of the upper bound (33) The proof of the upper bound is almost identical to the upper bound proof of Lemma 6. Once again, we will abuse notation by writing $g(\mathbf{y})$ instead of $g(\mathbf{y},\phi,\ldots,\phi)$ for any vector $\mathbf{y}$ of dimension $r < n$, where ϕ is some minimal-value element as defined in Section 2.

Analogous (but slightly different) than in the proof of Lemma 6, consider a deterministic vector $\mathbf{z} = (z_1, z_2, \ldots, z_{n-1})$ such that $g(\mathbf{z}) = \min\{ca^\tau_{\pi_\tau(S)}, g^{max}_{k-1}\}$, for a positive constant $c > 1$ whose value will be determined later. In choosing this vector, we will apply the definition of the extended diminishing returns condition so that for any $\mathbf{y}$ satisfying $g(\mathbf{y}) \leq g(\mathbf{z})$ and $x \geq 0$, Equation (21) is satisfied.

Let $S^* = \{v_1, v_2, \ldots, v_{n-1}\}$ be a set of (fictitious) items such that $X_{v_j} = z_j$ with probability 1 (the performance of each of these fictitious items is deterministic). Therefore, the performance of the set of items S^* is given by $u(S^*) = g(\mathbf{z}) = \min\{ca^\tau_{\pi_\tau(S)}, g^{max}_{k-1}\}$.

By definition, we know that $a^r_{\pi_r(S)} \leq a^\tau_{\pi_\tau(S)}$ for all r. Moreover, we can upper bound $u(S)$ as follows,

$$u(S) \leq u(S \cup S^*) \leq u(S^*) + \sum_{r=1}^{k}[u(S^* \cup \{\pi_r(S)\}) - u(S^*)]. \tag{34}$$

Let $X^{(1)}_{\pi_r(S)}, X^{(2)}_{\pi_r(S)}, \ldots, X^{(r)}_{\pi_r(S)}$ be independent random variables with distribution $P_{\pi_r(S)}$. Let $X = X^{(r)}_{\pi_r(S)}$ and $\mathbf{Y} = (X^{(1)}_{\pi_r(S)}, X^{(2)}_{\pi_r(S)}, \ldots, X^{(r-1)}_{\pi_r(S)})$. Note that

$$\begin{aligned} u(S^* \cup \{\pi_r(S)\}) - u(S^*) &= \mathbf{E}[g(\mathbf{z},X) - g(\mathbf{z})] \\ &= \mathbf{E}\left[g(\mathbf{z},X) - g(\mathbf{z}) \mid g(\mathbf{Y}) \leq g(\mathbf{z})\right] \\ &\overset{(a)}{\leq} \mathbf{E}\big[g(\mathbf{Y},X) - g(\mathbf{Y}) \mid g(\mathbf{Y}) \leq g(\mathbf{z})\big] \\ &\leq \frac{\mathbf{E}[g(\mathbf{Y},X) - g(\mathbf{Y})]}{\mathbf{Pr}[g(\mathbf{Y}) \leq g(\mathbf{z})]} \\ &\overset{(b)}{\leq} \frac{1}{\mathbf{Pr}[g(\mathbf{Y}) \leq g(\mathbf{z})]}\frac{a^r_{\pi_r(S)}}{r}. \end{aligned}$$

Inequality (a) follows from the extended diminishing returns property defined in Definition 1. Note that from our definition of $\mathbf{z}$, for any instantiation $\mathbf{Y}$ where $g(\mathbf{Y}) \leq g(\mathbf{z})$, extended diminishing returns tells us that $g(\mathbf{z},X) - g(\mathbf{z}) \leq g(\mathbf{Y},X) - g(\mathbf{Y})$ for all X. Taking the expectation over all $\mathbf{Y}, X$ conditional upon $g(\mathbf{Y}) \leq g(\mathbf{z})$ gives us (a).

Inequality (b) can be shown using only the definition of submodularity as can be seen via the below sequence of inequalities: suppose that $i = \pi_r(S)$.

$$\begin{aligned}\mathbf{E}[g(\mathbf{Y}, X) - g(\mathbf{Y})] &\leq \frac{1}{r}\sum_{s=0}^{r-1}\left(u(\{i^{(1)}, \ldots, i^{(s)}, i^{(r)}\}) - u(\{i^{(1)}, \ldots, i^{(s)}\})\right)\\ &= \frac{1}{r}\sum_{s=0}^{r-1}\left(u(\{i^{(1)}, \ldots, i^{(s)}, i^{(s+1)}\}) - u(\{i^{(1)}, \ldots, i^{(s)}\})\right)\\ &= \frac{1}{r}u(\{i^{(1)}, \ldots, i^{(r)}\})\\ &= \frac{a_i^r}{r} = \frac{a_{\pi_r(S)}^r}{r}.\end{aligned}$$

All that remains is to prove that $\mathbf{Pr}\big[g(\mathbf{Y}) \leq g(\mathbf{z})\big] \geq 1 - 1/c$.

Recall that $g(\mathbf{z}) = \min\{ca_{\pi_\tau(S)}^\tau, g_{k-1}^{max}\}$. Let us proceed by considering two cases depending on the value of $g(\mathbf{z})$. If $g(\mathbf{z}) = g_{k-1}^{max}$, then $\mathbf{Pr}\big[g(\mathbf{Y}) \leq g(\mathbf{z})\big] = 1$ trivially. This is because by definition g_{k-1}^{max} is the maximum value that the function can take on any vector of length $k-1$, and by monotonicity, any vector of size $r-1$ such as $\mathbf{Y}$ since $r \leq k$. On the other hand, when $g(\mathbf{z}) = ca_{\pi_\tau(S)}^\tau$, we can apply Markov's inequality and bound the desired probability, i.e.,

$$\mathbf{Pr}\left[g(\mathbf{Y}) \geq ca_{\pi_\tau(S)}^\tau\right] \leq \frac{\mathbf{E}\left[g(\mathbf{Y})\right]}{ca_{\pi_\tau(S)}^\tau} \leq \frac{1}{c}$$

where we used $\mathbf{E}[g(\mathbf{Y})] = a_{\pi_r(S)}^{r-1} \leq a_{\pi_r(S)}^r \leq a_{\pi_\tau(S)}^\tau$. Since $\mathbf{Pr}\left[g(\mathbf{Y}) \leq ca_{\pi_\tau(S)}^\tau\right] \geq 1 - \mathbf{Pr}\left[g(\mathbf{Y}) \geq ca_{\pi_\tau(S)}^\tau\right]$, it follows that $\mathbf{Pr}\left[g(\mathbf{Y}) \leq ca_{\pi_\tau(S)}^\tau\right] \geq 1 - 1/c$, as desired.

We have shown that $u(S^* \cup \{\pi_r(S)\}) - u(S^*) \leq (1 - 1/c)^{-1} a_{\pi_r(S)}^r / r$.

Combining with (34), we obtain

$$u(S) \leq ca_{\pi_\tau(S)}^\tau + \left(1 - \frac{1}{c}\right)^{-1}\left(\frac{a_{\pi_1(S)}^1}{1} + \frac{a_{\pi_2(S)}^2}{2} + \cdots + \frac{a_{\pi_k(S)}^k}{k}\right).$$

Applying Lemma 14 to $a_{\pi_\tau(S)}^\tau$, we obtain that

$$\begin{aligned}u(S) &\leq 2c\frac{1}{\tau}\sum_{r=1}^{\tau} a_{\pi_r(S)}^r + \left(1 - \frac{1}{c}\right)^{-1}\left(a_{\pi_1(S)}^1 + \frac{a_{\pi_2(S)}^2}{2} + \cdots + \frac{a_{\pi_k(S)}^k}{k}\right)\\ &\leq \left(2c + \left(1 - \frac{1}{c}\right)^{-1}\right)\left(a_{\pi_1(S)}^1 + \frac{a_{\pi_2(S)}^2}{2} + \cdots + \frac{a_{\pi_k(S)}^k}{k}\right)\end{aligned}$$

which completes the proof by taking $c = 2$.

G. Proof of Lemma 10

Before proving Lemma 10, we prove that our sketch function v_j as defined in (12) satisfies a simple monotonicity property. This property will be useful in the proof of Lemma 10.

Proposition 16. *Suppose v_j is a sketch function for a stochastic monotone submodular function u_j as defined in* (12) *and let $S = \{i_1, i_2, \ldots, i_{|S|}\} \subseteq N$ such that for all $r \in \{1, 2, \ldots, |S|\}$, $\pi_r(S, j) = i_r$. Then, the following inequalities hold for all $r \in \{1, 2, \ldots, |S|\}$:*

$$v_j(S) \geq v_j(S \setminus \{i_r\}) \geq v_j(S) - \frac{a_{i_r,j}^r}{r}.$$

Proof of Proposition 16 Fix some $r \in \{1, 2, \ldots, |S|\}$, and for all $t \neq r$, define ν_t such that $\pi_{\nu_t}(S \setminus \{i_r\}, j) = i_t$. That is, ν_t denotes item i_t's new 'rank' in the set $S \setminus \{i_r\}$. Note that $1 \leq \nu_t \leq |S| - 1$ and that:

$$v_j(S \setminus \{i_r\}) = \sum_{t \neq r} \frac{a^{\nu_t}_{i_t,j}}{\nu_t}. \tag{35}$$

We show via induction on t that for all $t \neq r$, $\nu_t \leq t$, i.e., removal of an item cannot hurt the 'rank' of another item. The claim is trivially true when $\nu_t = 1$ since $t \geq 1$. Consider an arbitrary $v_t > 1$, and suppose that the inductive hypothesis is true up to ν_{t-1}. Let us consider two cases: first, if $\nu_t < r$, then by definition $\pi_t(S, j) = \pi_t(S \setminus \{i_r\}, j) = i_t$ and so the inductive claim holds since $\nu_t = t$. Second, suppose that $\nu_t \geq r$: assume by contradiction that $\nu_t > t$. By the inductive hypothesis, it must be the case that $\pi_t(S \setminus \{i_r\}, j) \in \{i_t, i_{t+1}, \ldots, i_{|S|}\}$—indeed, for all $t' < t$, we have that $\nu_{t'} \leq t'$. However, we know by definition of π that for all $i \in \{i_{t+1}, \ldots, i_{|S|}\}$, it must be true that:

$$a^t_{i_t,j} > a^t_{i,j}.$$

Therefore, if $\nu_t > t$, then $\pi_t(S \setminus \{i_r\}, j) \in \{i_{t+1}, \ldots, i_{|S|}\}$—this would be a violation of the definition of π. Hence, the inductive hypothesis follows.

Now, in order to prove the proposition, we go back to (35),

$$\begin{aligned} v_j(S \setminus \{i_r\}) &= \sum_{t \neq r} \frac{a^{\nu_t}_{i_t,j}}{\nu_t} \\ &\leq \sum_{t \neq r} \frac{a^{\nu_t}_{i_{\nu_t},j}}{\nu_t} \\ &= \sum_{t=1}^{|S|-1} \frac{a^t_{i_t,j}}{t} \\ &\leq v(S). \end{aligned}$$

The crucial step above is the second inequality. There, we used the fact that $\nu_t \leq t$, and therefore, if $\nu_t = q$, then $a^q_{i_q,j} \geq a^q_{i_t,j}$ by definition of i_q for all $1 \leq q = \nu_t \leq |S| - 1$. The third inequality comes from changing the index from ν_t to t. In summary, we have shown that $v(S) \geq v(S \setminus \{i_r\})$ which is one half the proposition. In order to prove the other half, that is $v(S \setminus \{i_r\}) \geq v(S) - a^r_{i_r,j}/r$, we utilize the result from Lemma 13, namely that:

$$\frac{a^{\nu_t}_{i_t,j}}{\nu_t} \geq \frac{a^t_{i_t,j}}{t},$$

which is true because $\nu_t \leq t$. To conclude the proposition, we have that:

$$\begin{aligned} v(S) &= \sum_{t=1}^{|S|} \frac{a^t_{i_t,j}}{t} \\ &= \sum_{t \neq r} \frac{a^t_{i_t,j}}{t} + \frac{a^r_{i_r,j}}{r} \\ &\leq \sum_{t \neq r} \frac{a^{\nu_t}_{i_t,j}}{\nu_t} + \frac{a^r_{i_r,j}}{r} \\ &= v(S \setminus \{i_r\}) + \frac{a^r_{i_r,j}}{r}. \quad \square \end{aligned}$$

We are now ready to prove the main lemma.

(Proof of Lemma 10) We need to show that the greedy algorithm described in Algorithm 1 returns an assignment $\mathbf{S} = (S_1, S_2, \dots, S_m)$ that is a $\frac{1}{2}$-approximation to the optimum assignment $\mathbf{O} = (O_1, O_2, \dots, O_m)$ that maximizes $v(\mathbf{S}') = \sum_{j=1}^{m} v_j(S'_j)$ where the function v_j is as defined in (12). If the sketch function v_j is submodular, then one can simply apply the well-known result by Lehmann et al. (2006) for the submodular welfare maximization problem to show that the greedy algorithm yields the desired approximation factor. However, despite its simplicity, the sketch function v_j is not necessarily submodular, so we cannot directly use the existing proof for submodular welfare maximization as a black-box.

Before proving the result, we introduce some pertinent notation. Recall that our algorithm proceeds in rounds such that at each time step t, exactly one item $i \in A$ is added to a partition $j \in M$. Let $\mathbf{S}(t) = (S_1(t), S_2(t), \dots, S_m(t))$ denote the assignment at the end of time step t, i.e., $S_j(t)$ is the set of items assigned to partition $j \in M$ at the end of t unique assignments. For notational convenience, let $\mathbf{S}(0) = (\emptyset, \emptyset, \dots, \emptyset)$. Suppose that $\mathbf{O}(t) = (O_1(t), O_2(t), \dots, O_m(t))$ denotes the optimal (constrained) assignment such that for every $j \in M$, $S_j(t) \subseteq O_j(t)$, i.e., this assignment deviates from $\mathbf{S}$ only in the set of items that are unassigned at the end of time step t. Finally, suppose that at round $t+1$, if our algorithm assigns item $i \in N$ to partition $j \in M$, then the added welfare is $\Delta(t+1) := a_{i,j}^{|S_j(t)|+1}/(|S_j(t)|+1)$.

The basic idea behind our proof is similar to that of Theorem 12 in (Lehmann et al. 2006). Namely, we show that $v(\mathbf{O}(t)) \leq v(\mathbf{O}(t+1)) + \Delta(t+1)$ for all $t \in \{0, 1, \dots, \ell-1\}$, where ℓ is the total number of rounds the algorithm proceeds for. By cascading this argument, we can show the desired approximation guarantee, i.e.,

$$\begin{aligned} v(\mathbf{O}(0)) &\leq v(\mathbf{O}(1)) + \Delta(1) \qquad (36)\\ &\leq \cdots \\ &\leq v(\mathbf{O}(t)) + \sum_{r=1}^{t} \Delta(r) \\ &\leq \cdots \\ &\leq v(\mathbf{O}(\ell)) + \sum_{r=1}^{\ell} \Delta(r) \\ &= v(\mathbf{O}(\ell)) + v(\mathbf{S}(\ell)) \\ &= 2v(\mathbf{S}). \qquad (37) \end{aligned}$$

The first five equations above come from an application of the claimed inequality $v(\mathbf{O}(t)) \leq v(\mathbf{O}(t+1)) + \Delta(t+1)$ for all $t \in \{0, 1, \dots, \ell-1\}$. The penultimate and final equations follow from: (a) $\mathbf{O}(\ell) = \mathbf{S}(\ell) = \mathbf{S}$ by definition, and (b) the total welfare generated by the solution $\mathbf{S}$ is simply the sum of welfare added in each round, i.e., $\sum_{r=1}^{\ell} \Delta(r)$. Finally, this argument can be used to conclude the proof since $\mathbf{O}(0)$ is the same as the unconstrained optimum assignment $\mathbf{O}$ by definition.

All that remains for us is to prove the claim $v(\mathbf{O}(t)) \leq v(\mathbf{O}(t+1)) + \Delta(t+1)$ for all $t \in \{0, 1, \dots, \ell-1\}$. In (Lehmann et al. 2006), this claim followed from submodularity. However, since this is no longer a valid approach in our setting, we use a more subtle argument based on the monotonicity result from Proposition 16.

Suppose that in round $t+1$, our algorithm assigns item i to partition j and let $|S_j(t+1)| = r$ so that $\Delta(t+1) = a_{i,j}^{r}/r$. Moreover, suppose that in the constrained optimum solution $\mathbf{O}(t)$, item i is assigned to partition j' and integer parameter r' is such that $\pi_{r'}(O_{j'}(t), j') = i$. A crucial observation here is that $r' > |S_{j'}(t)|$. This is because Algorithm 1 greedily assigns the item with the maximum marginal benefit at each round and we know that item i was still unassigned at the end of round t. Consider the assignment $\mathbf{O}(t+1)$, we have that:

$$v(\mathbf{O}(t+1)) \geq v(\mathbf{O}(t)) + \big(v_j(O_j(t) \cup \{i\}) - v_j(O_j(t))\big) - \big(v_{j'}(O_{j'}(t)) - v_{j'}(O_{j'}(t) \setminus \{i\})\big). \tag{38}$$

Starting with the assignment $\mathbf{O}(t)$, if we move item i from partition j' to partition j, the resulting assignment has a welfare that is denoted by the right hand side of the above inequality. Now, since the resulting assignment also subsumes $\mathbf{S}(t+1)$, its welfare cannot be larger than $\mathbf{O}(t+1)$. Consider the term, $v_j(O_j(t) \cup \{i\}) - v_j(O_j(t))$ from the RHS of (38)—this is non-negative by the monotonicity argument laid out in Proposition 16. Similarly, consider the other term from the RHS, namely $v_{j'}(O_{j'}(t)) - v_{j'}(O_{j'}(t) \setminus \{i\})$—this is upper bounded by $a^{r'}_{i,j'}/r'$ as per Proposition 16 and our definition of r'. Further, according to Lemma 13, we have that:

$$\frac{a^{r'}_{i,j'}}{r'} \leq \frac{a^{|S_{j'}(t)|+1}_{i,j'}}{|S_{j'}(t)|+1},$$

since we proved earlier that $r' > S_{j'}(t)$. Putting all these ingredients together, we arrive upon the desired claim that $v(\mathbf{O}(t)) \leq v(\mathbf{O}(t+1)) + \Delta(t+1)$ for all $t \in \{0, 1, \dots, \ell - 1\}$:

$$\begin{aligned} v(\mathbf{O}(t+1)) &\geq v(\mathbf{O}(t)) + \big(v_j(O_j(t) \cup \{i\}) - v_j(O_j(t))\big) - \big(v_{j'}(O_{j'}(t)) - v_{j'}(O_{j'}(t) \setminus \{i\})\big) \\ &\geq v(\mathbf{O}(t)) + (0) - \frac{a^{r'}_{i,j'}}{r'} \qquad (39) \\ &\geq v(\mathbf{O}(t)) - \frac{a^{|S_{j'}(t)|+1}_{i,j'}}{|S_{j'}(t)|+1} \qquad (40) \\ &\geq v(\mathbf{O}(t)) - \frac{a^{r}_{i,j}}{r} \qquad (41) \\ &= v(\mathbf{O}(t)) - \Delta(t+1). \end{aligned}$$

Equation (39) is a product of the monotonicity claims from Proposition 16. Equation (40) is due to the fact that $r' > |S_{j'}(t)|$ and due to Lemma 13. Finally, the penultimate inequality (41) comes from the property of the greedy algorithm. At round $t+1$, since the greedy algorithm assigned item i to partition j as opposed to partition j', it must have been the case that $a^{|S_{j'}(t)|+1}_{i,j'}/(|S_{j'}(t)|+1) \leq a^{r}_{i,j}/r$. This concludes our proof. □

H. Sample Average Approximation Algorithms

H.1. NP-Hardness of Sample-Based Stochastic Optimization

We now present an example of a stochastic submodular optimization instance with a rather simple utility function where employing sample based algorithms may subsequently result in a discrete optimization problem that is NP-Hard. On the other hand, test score algorithms avoid the additional overhead brought about by solving secondary optimization problems. More concretely, consider the problem of maximizing a stochastic monotone submodular function subject to a cardinality constraint where $g(\mathbf{x}) = \max\{x_1, x_2, \dots, x_n\}$. For every $i \in N$, the distribution P_i is defined as follows: let X_i be a random variable such that $X_i = 1$ with probability p_i and $X_i = 0$ with probability $1 - p_i$ for some sufficiently small probabilities $(p_i)_{i \in N}$.

Consider the sample average approximation approach which first computes a collection of T independent sample vectors $(X_1^{(t)}, X_2^{(t)}, \ldots, X_n^{(t)})_{t=1}^{T}$, where $X_i^{(t)} \sim P_i$. For a given cardinality parameter k, the SAA method would look to compute a subset $S^* \subseteq N$ in order to maximize the number of 'covered indices' t, i.e., $\arg\max_{S \subseteq N} \sum_{t=1}^{T} \mathbb{1}\{\exists i \in S : X_i^{(t)} = 1\}$, where $\mathbb{1}$ is the indicator function that evaluates to one when the condition inside is true and is zero otherwise. However, this is equivalent to the well-studied maximum coverage problem which is known to be NP-Hard. Note that for the same instance, a test score algorithm based on replication scores would return the optimum solution with high probability since the test scores would be monotonically increasing in the probability p_i. In the following section, we delve deeper into the sample errors due to test score and SAA methods

H.2. Error Probability for Finite Samples

We discuss the use of sample averages for estimating test scores for the simple instance introduced in Example 1 and the numerical results provided in Section 5.1. Our goal is to characterize the probability of error in identifying an optimal set of items due to use of sample averages for approximating replication test scores for the aforementioned simple example. The simplicity of this example allows us to derive tight characterizations of the required number of samples for the probability of error to be within a prescribed bound. We also conduct a similar analysis for the sample averaging approach (SAA) that amounts to enumerating and estimating value of each feasible set of items, and compare with the test score based approach.

Recall that we consider a ground set of items N that consists of type-A and type-B items that reside in two disjoint nonempty sets A and B, respectively, such that $N = A \cup B$. For each $i \in A$, $X_i = a$ with probability 1, and for each $i \in B$, $X_i = b/p$ with probability p, and $X_i = 0$ otherwise, where $a, b > 0$ and $p \in (0, 1]$ are parameters. We assume that $b/p > a$ so that individual performance of a type-B item is larger than that of any type-A item conditional on the type-B item achieving performance b/p. We may think of type-B items as of high-risk, high-return items when p is small. We assume that for given k, $|A| \geq k$ and $|B| \geq k$.

We consider the best-shot utility function $u(S) = \mathbf{E}[\max\{x_i \mid i \in S\}]$, which want to maximize over sets $S \in 2^N$ of cardinality $|S| = k$. Clearly, we can distinguish $k+1$ equivalence cases for sets S with respect to the value of the utility function: class r defined by having r type-B items and $k - r$ type-A items, for $r \in \{0, 1, \ldots, k\}$. Let $C_{k,r}$ denote all sets of cardinality k that are of class r.

For each $S \in C_{k,r}$, we have

$$u(S) = \mathbf{E}[\max\{X_{(r)}, a\}]$$

where $X^{(r)}$ is the largest order statistic of individual performance of type-B items,

$$\mathbf{Pr}[X_{(r)} = b/p] = 1 - \mathbf{Pr}[X_{(r)} = 0] = 1 - (1-p)^r.$$

Indeed, we have

$$u(S) = a(1-p)^r + \frac{b}{p}(1 - (1-p)^r).$$

Since we assumed that $b/p > a$, we have that $u(S)$ is increasing in the class of set S, achieving the largest value for $r = k$, i.e. when all items are of type B.

In our analysis, we will make use of the well-known Hoeffding's inequality (Hoeffding 1963) to bound the probability of the event that a sum of independent random variables with bounded supports deviates from its expected value by more than a given amount.

Proposition 17 **(Hoeffding's inequality).** *Let $X_1, X_2, \ldots, X_T$ be independent random variables such that $X_i \in [\alpha_i, \beta_i]$ with probability 1 for all $i \in \{1,2,\ldots,T\}$. Then, for every $x \geq 0$,*

$$\mathbf{Pr}[X_1 + X_2 + \cdots + X_T - \mathbf{E}[X_1 + X_2 + \cdots + X_T] \geq x] \leq \exp\left(-\frac{2x^2T^2}{\sum_{i=1}^T (\beta_i - \alpha_i)^2}\right).$$

Test scores Consider sample average estimators of replication test scores defined as follows:

$$\hat{a}_i = \frac{1}{T}\sum_{t=1}^T \max\{X_i^{(1,t)}, X_i^{(2,t)}, \ldots, X_i^{(k,t)}\}$$

where $X_i^{(j,t)}$ are independent over i, j, and t and $X_i^{(j,t)}$ has distribution P_i. Indeed, by denoting $X_i^{((k),t)}$ the largest order statistic of P_i, we can write

$$\hat{a}_i = \frac{1}{T}\sum_{t=1}^T X_i^{((k),t)}.$$

Indeed, for our example, for every $i \in A$, we have $\hat{a}_i = a$. On the other hand, for every $i \in B$, we have that $X_i^{((k),t)}$ is equal to b/p with probability $1-(1-p)^k$ and is equal to 0 otherwise. Thus, for every $i \in B$, $a_i = \mathbf{E}[\hat{a}_i] = (b/p)(1-(1-p)^k)$. In what follows, we assume that $(b/p)(1-(1-p)^k) > a$, i.e. $\mathbf{E}[\hat{a}_i] < \mathbf{E}[\hat{a}_j]$ for every $i \in A$ and $j \in B$. In this case, in absence of estimation noise, the replication test score based algorithm correctly identifies an optimum set of items to be a set k type-B items. We declare an error event to occur if $\hat{a}_j < \hat{a}_i$ for some items $i \in A$ and $j \in B$, and denote with p_e the probability of this event, i.e. $p_e := \mathbf{Pr}[\cup_{i\in A, j\in B}\{\hat{a}_j < \hat{a}_i\}]$.

By the Hoeffding's inequality, for any type-A item i and type-B item j, we have

$$\mathbf{Pr}[\hat{a}_j < \hat{a}_i] = \mathbf{Pr}\left[\frac{1}{T}\sum_{t=1}^T X_i^{((k),t)} < a\right] \leq \exp(-2(1-(1-p)^k - ap/b)^2 T).$$

By the union bound, we have

$$p_e \leq |A||B| \exp\left(-2p^2\left((1-(1-p)^k)/p - a/b\right)^2 T\right).$$

Hence, for $p_e \leq \delta$ to hold, for given $\delta \in (0,1]$, it suffices that the total number of samples $m := nkT$ is such that

$$m \geq \frac{nk}{2p^2\left((1-(1-p)^k)/p - a/b\right)^2}\log\left(\frac{|A||B|}{\delta}\right). \tag{42}$$

Under given assumptions $|A| + |B| = n$ and $|A|, |B| \geq k$, we have $|A||B| \leq n^2/4$, so in (42), we can replace $\log(|A||B|/\delta)$ with $2\log(n/2) + \log(1/\delta)$ to obtain a sufficient number of samples. Note that $((1-(1-p)^k)/p - a/b)^2 = (k - a/b)^2(1+o(1))$ for small p. Hence, we have $m = \Omega(1/p^2)$.

SAA approach Consider now a stochastic average approximation method that amounts to enumerating all feasible sets and then choosing the one that has the best estimated value: for each $S \subseteq N$ such that $|S| = k$, estimating $u(S)$ with the sample average $\hat{u}(S)$ defined as

$$\hat{u}(S) = \frac{1}{T}\sum_{t=1}^T \max\{X_i^{(t)} \mid i \in S\}$$

where $X_i^{(t)}$ are independent random variables over i and t and $X_i^{(t)} \sim P_i$ for all $S \in 2^N$ and $t \in \{1, 2 \ldots, T\}$.

For every class-0 set S, whose all elements are of type A, we have $\hat{u}(S) = a$ with probability 1. For every class-r set S, with $1 \leq r < k$, we have $\hat{u}(S) \geq a$. For every class-r set S, with $0 \leq r \leq k$, we have

$$\hat{u}(S) = a\left(1 - \frac{X_S}{T}\right) + \frac{b}{p}\frac{X_S}{T}$$

where $X_S \sim \mathrm{Bin}(T, 1-(1-p)^r)$.

Comparing $\hat{u}(S) > \hat{u}(S')$ for any two sets S and S' is equivalent to $X_S > X_{S'}$. By the Hoeffding's inequality, for an two sets S and S' such that $\mathbf{E}[X_S] > \mathbf{E}[X_{S'}]$, we have

$$\mathbf{Pr}[X_S \leq X_{S'}] \leq \exp\left(-\frac{1}{2}(\mathbf{E}[X_S] - \mathbf{E}[X_{S'}])^2 T\right). \tag{43}$$

We declare an error event to occur if $\hat{u}(S) < \hat{u}(S')$ for every class k set S and some class $r < k$ set S' and denote with p_e the probability of this event. Then, by the union bound, we have

$$\begin{aligned}
p_e &= \mathbf{Pr}[X_S < X_{S'} \text{ for every } S \in C_{k,k} \text{ and some } S' \in \cup_{0 \leq r < k} C_{k,r}] \\
&\leq \mathbf{Pr}[\cup_{S' \in \cup_{0 \leq r < k} C_{k,r}} \{X_{S_k} < X_{S'}\}] \\
&\leq \sum_{r=0}^{k-1} |C_{k,r}| \mathbf{Pr}[X_{S_k} \leq X_{S_r}] \\
&\leq \left(\sum_{r=0}^{k-1} |C_{k,r}|\right) \mathbf{Pr}[X_{S_k} \leq X_{S_{k-1}}] \\
&= \left(\binom{n}{k} - \binom{|B|}{k}\right) \mathbf{Pr}[X_{S_k} \leq X_{S_{k-1}}]
\end{aligned}$$

where S_i denotes an arbitrarily fixed set in $C_{k,i}$.

Combining with (43), we have

$$p_e \leq \left(\binom{n}{k} - \binom{|B|}{k}\right) \exp\left(-\frac{1}{2} p^2 (1-p)^{2(k-1)} T\right). \tag{44}$$

Note that the error exponent in (44) is due to discriminating a class k set from a class $k-1$ set. In order to have $p_e \leq \delta$, for given $\delta \in (0,1]$, it suffices for the total number of samples $m := nT$ to be such that

$$m \geq \frac{2n}{p^2(1-p)^{2(k-1)}} \log\left(\frac{\binom{n}{k} - \binom{|B|}{k}}{\delta}\right). \tag{45}$$

Note that in (45) we can replace $\binom{n}{k} - \binom{|B|}{k}$ with $\binom{n}{k}$, which is tight for $|B| = \Theta(k)$. Furthermore, we can use the well known inequalities $k\left(\log\left(\frac{n}{k}\right)\right) \leq \log\left(\binom{n}{k}\right) \leq k\left(\log\left(\frac{n}{k}\right) + 1\right)$. Thus, the logarithmic term in (45) contributes a factor of k to the sufficient number of samples. Note also that $m = \Omega(1/p^2)$.

Summary The analysis of the estimation error for the SAA approach requires to consider discrimination of a set with all type-B items and a set that has at least one type-A item. On the other hand, for the approach based on using replication test scores, we only need to consider discrimination of a set with all type-B items and a set with all type-A items. For both approaches, we obtain that the error exponent scales as $\Theta(p^2)$ for small p. The SAA approach can require a larger number of samples than the replication test score approach, which is demonstrated by numerical results in Section 5.1.

I. Proof of Proposition 11

Let $X_1, X_2, \dots, X_n$ be independent random variables with distributions $P_1, P_2, \dots, P_n$, respectively, and let $\mathbf{X} := (X_1, X_2, \dots, X_n)$. Without loss of generality, assume that items are enumerated in decreasing order of mean test scores, i.e. $\mathbf{E}[X_1] \geq \mathbf{E}[X_2] \geq \cdots \geq \mathbf{E}[X_n]$. Let $S = \{i_1, i_2, \dots, i_k\}$ be an arbitrary subset of items in N. Then, we have

$$\begin{aligned} u(S) &= \mathbf{E}[g(M_S(\mathbf{X}))] \\ &= \mathbf{E}[[g(M_S(\mathbf{X})) - g(M_{S\setminus\{i_k\}}(\mathbf{X}))] + [g(M_{S\setminus\{i_k\}}(\mathbf{X})) - g(M_{S\setminus\{i_{k-1},i_k\}}(\mathbf{X}))] + \cdots + [g(M_{\{i_1\}}(\mathbf{X})) - g(\phi,\dots,\phi)]] \\ &= [u(S) - u(S\setminus\{i_k\})] + [u(S\setminus\{i_k\}) - u(S\setminus\{i_{k-1},i_k\})] + \cdots + [u(\{i_1\}) - u(\emptyset)] \\ &\leq u(\{i_k\}) + u(\{i_{k-1}\}) + \cdots + u(\{i_1\}) \\ &= \sum_{i\in S} \mathbf{E}[X_i] \\ &\leq \sum_{i=1}^{k} \mathbf{E}[X_i] \end{aligned} \tag{46}$$

where the first inequality follows by the submodularity of function u, the second inequality is by the assumption that items are enumerated in decreasing order of their mean test scores.

By Jensen's inequality, for every $(x_1, x_2, \dots, x_k) \in \mathbf{R}_+^k$, we have

$$\frac{1}{k}\sum_{i=1}^{k} x_i = \frac{1}{k}\sum_{i=1}^{k} (x_i^r)^{1/r} \leq \left(\frac{1}{k}\sum_{i=1}^{k} x_i^r\right)^{1/r}.$$

Hence, we have

$$\sum_{i=1}^{k} \mathbf{E}[X_i] \leq k^{1-1/r}\mathbf{E}\left[\left(\sum_{i=1}^{k} X_i^r\right)^{1/r}\right]. \tag{47}$$

From (46) and (47), for every $S \subseteq N$ such that $|S| = k$,

$$u(M) = \mathbf{E}\left[\left(\sum_{i=1}^{k} X_i^r\right)^{1/r}\right] \geq \frac{1}{k^{1-1/r}}\mathbf{E}\left[\left(\sum_{i\in S} X_i^r\right)^{1/r}\right] = \frac{1}{k^{1-1/r}}u(S).$$

The tightness can be established as follows. Let N consist of two disjoint subsets of items M and R, where M is a set of k items whose each individual performance is of value $1+\epsilon$ with probability 1, for parameter $\epsilon > 0$, and R is a set of k items whose each individual performance is of value a with probability $1/a$ and of value 0 otherwise, for parameter $a \geq 1$. Then, we note that

$$u(M) = k^{1/r}(1+\epsilon)$$

and

$$\begin{aligned} u(\mathrm{OPT}) \geq u(R) &= \mathbf{E}\left[\left(\sum_{i\in R} X_i^r\right)^{1/r}\right] \\ &\geq a\mathbf{Pr}\left[\sum_{i\in R} X_i > 0\right] \\ &= a\left(1 - \left(1 - \frac{1}{a}\right)^k\right) \\ &\geq a\left(1 - e^{-k/a}\right). \end{aligned}$$

Hence, it follows that

$$\frac{u(M)}{u(\mathrm{OPT})} \leq (1+\epsilon)\frac{1}{k^{1-1/r}}\frac{k/a}{1-e^{-k/a}}.$$

The tightness claim follows by taking a such that $k = o(a)$, in which case $(k/a)/(1-e^{-k/a}) = 1+o(1)$.

J. Proof of Proposition 12

Proof of Claim (a) If k is a constant, then there is no r satisfying both conditions $r = o(1)$ and $r > 1$. Hence, it suffices to consider $k = \omega(1)$ and show that the following statement holds: for any given $\theta > 0$, there exists an instance for which greedy selection in decreasing order of quantile test scores cannot give a constant-factor approximation.

Consider the distributions of random variables X_i defined as follows:

1. Let X_i be equal to a with probability 1 for $1 \leq i \leq k$. For each of these items, the quantile test score is equal to a and the replication score is equal to $ak^{1/r}$.
2. Let X_i be equal to 0 with probability $1-1/n$, and equal to $b\theta n/k$ with probability $1/n$ for $k+1 \leq i \leq 2k$. Note that in the limit as n grows large, each of these items has quantile test score of value b and replication score of value $b\theta$.
3. Let X_i be equal to 0 with probability $1-\theta/k$ and equal to c with probability θ/k for $2k+1 \leq i \leq 3k$. For each of these items, the quantile test score is equal to c and the replication test score is less than or equal to $c\theta^{1/r}$.
4. Let X_i be equal to 0 for $3k+1 \leq i \leq n$.

If θ is a constant, i.e., $\theta = O(1)$, we can easily check that greedy selection in decreasing order of quantile test scores cannot give a constant-factor approximation with $a = b = 1$ and $c = 2$. Under this condition, the selected set of items is $\{2k+1,\ldots,3k\}$. However, we have

$$\begin{aligned}
\frac{\mathbf{E}\left[\left(\sum_{i=2k+1}^{3k} X_i^r\right)^{1/r}\right]}{\mathbf{E}\left[\left(\sum_{i=1}^{k} X_i^r\right)^{1/r}\right]} &= \frac{\mathbf{E}\left[\left(\sum_{i=2k+1}^{3k} X_i^r\right)^{1/r}\right]}{k^{1/r}} \\
&\leq \frac{\left(\sum_{i=2k+1}^{3k} \mathbf{E}\left[X_i^r\right]\right)^{1/r}}{k^{1/r}} \\
&= 2\left(\frac{\theta}{k}\right)^{1/r} \\
&= o(1),
\end{aligned}$$

which is because $k = \omega(1)$, $\theta = O(1)$, and $r = o(\log(k))$.

Since $r > 1$, if θ goes to infinity as n goes to infinity, i.e. for $\theta = \omega(1)$, we have

$$\begin{aligned}
\frac{\mathbf{E}\left[\left(\sum_{i=2k+1}^{3k} X_i^r\right)^{1/r}\right]}{\mathbf{E}\left[\left(\sum_{i=k+1}^{2k} X_i^r\right)^{1/r}\right]} &\leq \frac{\left(\sum_{i=2k+1}^{3k} \mathbf{E}\left[X_i^r\right]\right)^{1/r}}{\theta} \\
&= 2\theta^{(1-r)/r} \\
&= o(1).
\end{aligned}$$

Therefore, the greedy selection in decreasing order of quantile test scores has a vanishing utility compared to the optimal value.

Proof of Claim (b) Let $T(X,S)$ be a subset of S such that $i \in T(X,S)$ if, and only if, $X_i \geq P_i^{-1}(1-1/k)$, for $i \in S$. Let $a_{\max} = \max_{i\in S} a_i$ and $a_{\min} = \min_{i\in S} a_i$. We will show that there exist constants q and p such that

$$pa_{\min} \leq \mathbf{E}\left[\left(\sum_{i\in S} X_i^r\right)^{1/r}\right] \leq qa_{\max}.$$

Since $(x+y)^{1/r} \leq x^{1/r} + y^{1/r}$ for all $x,y \geq 0$ and $r > 1$, we have

$$\begin{aligned}
\mathbf{E}\left[\left(\sum_{i\in S} X_i^r\right)^{1/r}\right] &= \mathbf{E}\left[\left(\sum_{i\in T(X,S)} X_i^r + \sum_{i\in S\setminus T(X,S)} X_i^r\right)^{1/r}\right] \\
&\leq \mathbf{E}\left[\left(\sum_{i\in T(X,S)} X_i^r\right)^{1/r} + \left(\sum_{i\in S\setminus T(X,S)} X_i^r\right)^{1/r}\right] \\
&\leq \mathbf{E}\left[\sum_{i\in T(X,S)} X_i + \left(\sum_{i\in S\setminus T(X,S)} X_i^r\right)^{1/r}\right] \\
&\leq \mathbf{E}\left[\sum_{i\in T(X,S)} X_i + \left(\sum_{i\in S\setminus T(X,S)} (a_{\max})^r\right)^{1/r}\right] \\
&\leq \left(\mathbf{E}\left[|T(X,S)|\right] + k^{1/r}\right) a_{\max} \\
&= (1+k^{1/r})a_{\max}.
\end{aligned}$$

By the Minkowski inequality, $\left(\sum_{i\in A}\mathbf{E}\left[X_i\right]^p\right)^{1/p} \leq \mathbf{E}\left[\left(\sum_{i\in A} X_i^p\right)^{1/p}\right]$ for all $A \subseteq S$. Thus, we have

$$\begin{aligned}
\mathbf{E}\left[\left(\sum_{i\in S} X_i^p\right)^{1/p}\right] &= \mathbf{E}\left[\left(\sum_{i\in T(X,S)} X_i^p + \sum_{i\in S\setminus T(X,S)} X_i^p\right)^{1/p}\right] \\
&\geq \mathbf{E}\left[\left(\sum_{i\in T(X,S)} X_i^p\right)^{1/p}\right] \\
&= \sum_{A\subseteq S} \mathbf{Pr}\{T(X,S)=A\}\mathbf{E}\left[\left(\sum_{i\in A} X_i^p\right)^{1/p}\middle| T(X,S)=A\right] \\
&\geq \sum_{A\subseteq S} \mathbf{Pr}[T(X,S)=A]\left(\sum_{i\in A}\mathbf{E}[X_i | i \in T(X,S)]^p\right)^{1/p} \\
&\geq \sum_{A\subseteq S} \mathbf{Pr}\{T(X,S)=A\}|A|^{1/p}a_{\min} \\
&\geq \left(1-(1-1/k)^k\right) a_{\min} \\
&\geq (1-1/e)a_{\min}.
\end{aligned}$$

Therefore, the greedy selection in decreasing order of quantile test scores gives a constant-factor approximation of the optimal value.

K. Sketch Functions Based Lower Bounds for Submodular Welfare Maximization

In Section 4 we established a logarithmic approximation guarantee for the stochastic monotone submodular welfare maximization problem by designing a strong sketch function $v(S)$ that only used test scores to approximate the original utility function $u(S)$. In particular, in Lemma 9, we showed that there exist positive constants c_1 and c_2 such that for every $S \subseteq N$ of cardinality k, we have

$$c_2 \frac{v(S)}{\log(k)} \leq u(S) \leq c_1 v(S). \tag{48}$$

This allowed us to approximate the optimum solution to the submodular welfare maximization problem up to a $\Omega(\frac{1}{\log(k)})$-factor by selecting an assignment that optimizes the aggregate sketch function, i.e., $\sum_{j=1}^{m} v_j(S_j)$. As can be gleaned from Lemma 8, the approximation factor obtained using such an approach is closely tied to the fidelity of the sketch function, i.e., how well it approximates the original utility function.

This raises a natural question: *is it possible to obtain a constant factor approximation for this problem by identifying a better strong sketch function*? Note that by definition, such a sketch function would have to approximate the utility function $u(S)$ up to a constant factor at every input set S. In this section, we present negative examples indicating that such a strong sketch function, depending only on replication scores, may not exist. Taking into account this evidence, it seems plausible that if at all a constant factor approximation exists for this problem, then it would involve techniques other than sketching which are beyond the scope of this paper. While it would certainly be interesting to identify such techniques, we believe the sketch functions presented in this work provide additional value due to their broader applicability. For example, if the decision maker seeks to maximize a different objective or wishes to (approximately) evaluate the function value at some arbitrary input when oracle queries are costly, one could utilize score based sketch functions to achieve these tasks. For this reason, function sketching—even for submodular functions—has received considerable attention in the literature in the context of other problems (Goemans et al. 2009, Iyer and Bilmes 2013). Given that our objective is to illustrate how test score based methods can be applied to yield good solutions for different selection and assignment problems, our usage of sketch functions is rather pertinent.

Recall that for any given set S of cardinality k, the sketch function that was proposed in Section 4 used as inputs the following test scores $a^1_{\pi_1(S)}, a^2_{\pi_2(S)}, \dots, a^k_{\pi_k(S)}$, whose definitions can be found in (11). Note that in this section, we ignore the index j that is present in (11) and use $\pi(S)$ instead of $\pi(S, j)$ since we only consider a single utility function $u(S)$.

Our first result highlights the difficulty of using test scores to obtain strong sketches by showing that there exist two separate instances with the same objective function where the set of test scores is identical but the function value can be greatly different.

Claim 18. *For any set $S = \{1, 2, \dots, k\}$ of items and the value function $g(\mathbf{x}) = \max\{x_i \mid i \in S\}$, there exist two product-form distributions $P(x_1, x_2, \dots, x_k) = P_1(x_1)P_2(x_2)\cdots P_k(x_k)$ and $\tilde{P}(x_1, x_2, \dots, x_k) = \tilde{P}_1(x_1)\tilde{P}_2(x_2)\cdots\tilde{P}_k(x_k)$ such that*

1. *The test scores are identical for the two distributions:*

$$a^r_{\pi_r(S)} = \tilde{a}^r_{\tilde{\pi}_r(S)} \text{ for all } r \in \{1, 2, \dots, k\},$$

where $a^r_{\pi_r(S)}$ and $\tilde{a}^r_{\tilde{\pi}_r(S)}$ are the test scores with respect to distributions P and $\tilde{P}$ respectively, and

2. *The expected utility value for the first instance is a logarithmic factor larger than for the second instance:*

$$\mathbf{E}_{\mathbf{X}\sim P}\left[\max\{X_i \mid i \in S\}\right] = O(\log(k))\mathbf{E}_{\mathbf{X}\sim \tilde{P}}\left[\max\{X_i \mid i \in S\}\right].$$

In simple terms, this claim states that there are multiple instances across which the test scores are identical but the utility function value can differ by a logarithmic factor. As a consequence, *any sketch function* that relies on these test scores cannot approximate the original function by a factor better than $O(\log(k))$ for both these instances. This is due to the fact that the sketch function $v(S)$ depends only on test scores and must output an identical value for both these instances. To conclude, algorithms that optimize replication score based sketch functions may end up selecting or assigning sets that are sub-optimal by a logarithmic factor and thus this approach cannot be used to get an approximation factor better than $\Omega(\frac{1}{\log(k)})$ for submodular welfare maximization. Following the proof of the above claim, we depict a concrete example of an instance of stochastic submodular welfare maximization, where optimizing the sketch function leads to an assignment that is only a $\Theta(\frac{1}{\log(k)})$-approximation to the true optimum.

Proof of Claim 18 Consider a set of items $S = \{1, 2, \dots, k\}$. Our goal is to define two instances in terms of their item performance distributions $P_1(x), P_2(x), \dots, P_k(x)$ and $\tilde{P}_1(x), \tilde{P}_2(x), \dots, \tilde{P}_k(x)$ such that the conditions specified in the claim are satisfied for the set S. We define these distributions as follows:

$$\mathbf{Pr}_{X_i \sim P_i}[X_i = 1] = 1 - \mathbf{Pr}_{X_i \sim P_i}[X_i = 0] = q_i = 1 - \left(1 - \frac{1}{\log(k)}\right)^{\frac{1}{i}} \tag{49}$$

and

$$\mathbf{Pr}_{X_i \sim \tilde{P}_i}\left[X_i = \frac{1}{\log(k)}\right] = 1. \tag{50}$$

Next, we introduce the test scores for the two instances in the same manner as done in Section 4. Specifically, for all $i \in S$ and $r \in \{1, 2 \dots, k\}$, we have $a_i^r = \mathbf{E}_{X_i^{(j)} \sim P_i}[\max\{X_i^{(1)}, X_i^{(2)}, \dots, X_i^{(r)}\}]$ and $\tilde{a}_i^r = \mathbf{E}_{X_i^{(j)} \sim \tilde{P}_i}[\max\{X_i^{(1)}, X_i^{(2)}, \dots, X_i^{(r)}\}]$. Similar to our notation in Section 4, we define $\pi_r(S)$ and $\tilde{\pi}_r(S)$ in a recursive fashion as follows

$$\pi_1(S) = \arg\max_{i \in S} a_i^1; \qquad \pi_r(S) = \arg\max_{i \in S \setminus \{\pi_1(S), \dots, \pi_{r-1}(S)\}} a_i^r \quad \text{for } 1 < r \le k$$
$$\tilde{\pi}_1(S) = \arg\max_{i \in S} \tilde{a}_i^1; \qquad \tilde{\pi}_r(S) = \arg\max_{i \in S \setminus \{\tilde{\pi}_1(S), \dots, \tilde{\pi}_{r-1}(S)\}} \tilde{a}_i^r \quad \text{for } 1 < r \le k.$$

Informally, $(\pi_1(S), \dots, \pi_k(S))$ and $(\tilde{\pi}_1(S), \dots, \tilde{\pi}_k(S))$ are permutations of the items in the set S such that the r–th element maximizes the test scores a_i^r, $\tilde{a}_i^r$ respectively among the items not assigned to previous indices.

The test score equality is easy to prove. First we note that as per the definitions, $\pi_r(S) = r$ for all $1 \le r \le k$. Indeed, for the first instance, we have that $q_1 \ge q_2 \ge \dots \ge q_k$. Next, for any given r, we have that $a_{\pi_r(S)}^r$ is the expected performance of a group composed of r copies of item $\pi_r(S)$ and therefore,

$$a_{\pi_r(S)}^r = 1 - (1 - q_{\pi_r(S)})^r = 1 - \left(1 - \frac{1}{\log(k)}\right) = \frac{1}{\log(k)}.$$

In the case of the second instance, the items are identical and so, the permutation is irrelevant. For all r, we have that $\tilde{a}_{\tilde{\pi}_r(S)}^r = \frac{1}{\log(k)}$. Clearly, the test scores are identical.

In the case of the first instance under distribution P, the objective function can be calculated using the following steps:

$$\mathbf{E}_{\mathbf{X} \sim P}[\max\{X_i \mid i \in S\}] = 1 - \prod_{r=1}^{n}(1 - q_r)$$

$$
\begin{aligned}
&= 1 - \prod_{r=1}^{n} \left(1 - \frac{1}{\log(k)}\right)^{\frac{1}{r}} \\
&= 1 - \left(1 - \frac{1}{\log(k)}\right)^{\sum_{r=1}^{k} \frac{1}{r}} \\
&\geq 1 - \left(1 - \frac{1}{\log(k)}\right)^{\log(k)} \\
&\geq 1 - \frac{1}{e}.
\end{aligned}
$$

In the penultimate inequality, we used the fact that $\sum_{r=1}^{k} \frac{1}{r} \geq \log(k)$ and in the final step, we applied the inequality $(1 - \frac{1}{x})^x \leq \frac{1}{e}$ for all $x > 1$.

Now, evaluating the objective function for the second instance under distribution $\tilde{P}$ is rather straightforward; since all of the items are deterministic and identical, we have that:

$$\mathbf{E}_{\mathbf{X}\sim\tilde{P}}[\max\{X_i \mid i \in S\}] = \frac{1}{\log(k)}.$$

This completes the proof. □

As a corollary, we can now infer that both the lower bound and the upper bound in (48) are tight up to a constant factor for the same utility function. Note that the tightness of the bounds holds not just for the sketch functions devised in Section 4 but for every possible sketch function that is based solely on replication test scores. To see this, assume that the ground set N contains two disjoint sets S_1 and S_2 of items of cardinality n such that $S_1 = \{i_1, i_2, \dots, i_n\}$ and $\mathbf{Pr}[X_{i_j} = 1] = 1 - \mathbf{Pr}[X_{i_j} = 0] = q_j$ where $q_j = 1 - (1 - 1/\log(n))^{1/j}$ and for every $i \in S_2$, $X_i = 1/\log(n)$ with probability 1. Suppose that for any strong sketch function $v(S)$ that depends on $a^1_{\pi_1(S)}, a^2_{\pi_2(S)}, \dots, a^{|S|}_{\pi_1(S)}$, we have that:

$$u(S_1) \leq c'_1 v(S_1) \text{ and } u(S_2) \geq c'_2 v(S_2).$$

Since $v(S_1) = v(S_2)$ and $u(S_1) \geq (1 - \frac{1}{e})\log(n)u(S_2)$ by Claim 18, this implies that $\frac{c'_1}{c'_2} \geq (1 - \frac{1}{e})\log(n)$. In other words, if $v(S)$ is *any* strong sketch function based on replication scores that satisfies $c'_1 v(S) \geq u(S) \geq c'_2 v(S)$ for all S and c'_1, c'_2 are tight, then it is necessary that $\frac{c'_1}{c'_2} = \Omega(\log(n))$.

We now tie this back to stochastic submodular welfare maximization by illustrating how the gap in replication score based sketch functions implies the non-existence of a constant factor approximation algorithm. First, suppose that $\mathrm{OPT} = (\mathrm{OPT}_1, \dots, \mathrm{OPT}_m)$ denotes the optimal solution to an instance of the stochastic submodular welfare maximization problem and let $(S_1, \dots, S_m)$ be the assignment that maximizes the sum of the sketch functions. Then, by Lemma 8, it follows that

$$\sum_{j=1}^{m} u_j(S_j) \overset{(a)}{\geq} \sum_{j=1}^{m} c'_2 v_j(S_j) \geq \sum_{j=1}^{m} c'_2 v_j(\mathrm{OPT}_j) \overset{(b)}{\geq} \frac{c'_2}{c'_1} \sum_{j=1}^{m} u_j(\mathrm{OPT}_j).$$

Since both inequalities (a) and (b) can hold with equality and $\frac{c'_1}{c'_2} = \Omega(\log(n))$, one cannot use a sketch function $v(S)$ that depends only on replication scores to obtain a constant approximation factor. To illustrate this point in a more concrete sense, we briefly describe an example where optimizing a replication score based sketch function can lead to a sub-optimal solution for the welfare maximization problem.

Negative Example for Stochastic Submodular Welfare Maximization: Consider a ground set N composed of $2k$ items and the case of two partitions, i.e., $|M| = 2$. The $2k$ items are defined based on the two instances that we constructed in the proof of Claim 18. Specifically, $N = S_1 \cup S_2$ where S_1 and S_2 are disjoint sets of cardinality k such that for every type-1 item $i \in S_1$ its performance is 1 with probability q_i and 0 otherwise (where q_i is defined in (49)), and for every type-2 item $i \in S_2$, its performance is $1/\log(k)$ with probability 1. The first partition has utility function $u_1(S) = \mathbf{E}[\max\{X_i \mid i \in S\}]$ and the second partition has utility function $u_2(S) = \mathbf{E}[\epsilon \sum_{i \in S} X_i]$ for a sufficiently small constant $\epsilon > 0$. The two partitions are of cardinalities $k_1 = k_2 = k$.

It is easy to see that by assigning the type-1 items to the first partition, we can achieve a welfare that is at least of value $1 - 1/e$. Now, let us consider an algorithm that optimizes an arbitrary sketch function $v(S)$ which only depends on the scores $a^1_{\pi_1(S)}, a^2_{\pi_2(S)}, \ldots, a^{|S|}_{\pi_{|S|}(S)}$ for any given set S. Without loss of generality, suppose that any ties while deciding on the permutation $\pi_r(S)$ are broken in favor of type-1 items. We claim that the sketch function based algorithm would always assign the type-2 items to the first partition and achieve a welfare that is $\Theta(1/\log(k))$. To see why, first observe that from Claim 18, the sketch function should have the same value for $v_1(S_1)$ and $v_1(S_2)$ since the underlying test scores are identical. Second, for any $1 \leq r \leq k$, if at all the (optimal) algorithm assigns set S comprising of $1 \leq r \leq k$ items of type 1 to the first partition, this would be the r items from set S_1 that have the highest probabilities of success. However, these r items would appear first in the permutation $\pi(S, 1)$. This would lead to all test scores being equal, i.e., $a^r_{\pi_r(S,1)} = 1/\log(k)$ for all $1 \leq r \leq |S|$. Therefore, $v_1(S) = v_1(S_1) = v_1(S_2)$. Since the algorithm cannot distinguish between the sets S_1 and S_2 in the context of the first partition but the type-1 items have larger test scores under the second partition, the sum-of-sketch-functions is maximized by assigning S_1 to partition 2 and S_2 to partition 1. However, this only results in a social welfare of value $\Theta(1/\log(k))$.

Good Test Scores: Claim 18 implies that strong sketch functions based on replication scores cannot be used to guarantee a constant-factor approximation for submodular welfare maximization. Given this (partial) impossibility result, it is natural to consider whether the theory of good test scores developed for submodular function maximization could be used to obtain better algorithms for the more general problem. In this section, we study this question and provide negative examples that highlight why the good test scores identified in Lemma 6 cannot be leveraged to obtain constant-factor approximations for the welfare maximization problem.[8]

First, let us recall the upper and lower bounds developed in the proof of Lemma 6, namely, for all $S \subseteq N$,

$$\left(1 - \frac{1}{e}\right) \underline{v}(S) \leq u(S) \leq 4\bar{v}(S)$$

where $\underline{v}(S) = \min\{a^{|S|}_i \mid i \in S\}$ and $\bar{v}(S) = \max\{a^{|S|}_i \mid i \in S\}$. In simple terms, the largest and smallest test scores of the items serve as upper and lower bounds for the function value respectively, up to a constant factor. A natural possibility is to examine whether computing allocations (assignments of items to partitions) that maximize the sum of (either) the maximum or minimum test scores would come close to the optimal allocation over all feasible assignments. However, as we show next, such approaches can lead to highly sub-optimal assignments even for simple instances.

[8] These are based on the (p, q)-sketches from Definition 3.

Claim 19. *There exist instances of the submodular welfare maximization problem where the allocations that maximize $\underline{v}(S_1, S_2, \dots, S_m) = \sum_{j=1}^m \min\{a_i^{|S_j|} \mid i \in S_j\}$ and $\bar{v}(S_1, S_2, \dots, S_m) = \sum_{j=1}^m \max\{a_i^{|S_j|} \mid i \in S_j\}$, respectively, are $O(1/\sqrt{n})$-approximations to the welfare-maximizing allocations for the corresponding instances.*

Proof of Claim 19 First we show that maximizing the sum of the minimum test scores results in welfare that is only a $1/\sqrt{n}$-approximation to the optimum.

Example 1: Consider a problem instance with $n = r^2$ items and $m = r$ partitions with each partition having a cardinality constraint with $k_j = r$ for all j. All items are assumed to exhibit deterministic performance: r items (referred to as heavy items) have performance of value 1, i.e., $X_i = 1$ with probability 1, while the remaining items have performance of zero value. Assume that objective functions are best-shot functions, i.e., $g_j(S) = \max\{x_i \mid i \in S\}$ for each partition j.

The optimum solution for the given problem instance is when each of the heavy items is assigned to a different partition, leading to the welfare of value r. On the contrary, the solution that maximizes $\underline{v}(S_1, S_2, \dots, S_m) = \sum_{j=1}^m \min\{a_i^{|S_j|} \mid i \in S_j\}$ assigns all heavy items to same partition, which yields a welfare of value 1. Hence, this approach achieves a welfare that is $1/\sqrt{n}$ factor of the optimum, which can be made arbitrarily small by choosing large enough number of items n.

Next, we present an example where maximizing the sum of the maximum test scores also results in a poor approximation.

Example 2: Consider a problem instance with $n = 2r$ items and $m = r + 1$ partitions, where partition 1 has a cardinality constraint with $k_1 = r$, and each partition $1 < j \leq m$ has $k_j = 1$. All items are assumed to have deterministic performance once again: one heavy item with performance of value $\sqrt{r}$, $r - 1$ medium items with performance of value of 1, and, finally, the remaining items with zero-valued performance. Assume that value functions are $g_1(\mathbf{x}) = \sum_{i=1}^{k_1} x_i$ and $g_j(\mathbf{x}) = (1/\sqrt{r}) \max\{x_i \mid i = 1, 2, \dots, k_j\}$ for partitions $1 < j \leq m$.

The optimum solution assigns all items to partition 1, which yields a welfare of value $r + \sqrt{r} - 1$, whereas the algorithm assigns the heavy item to partition 1 and the medium items spread across, which yields a welfare of value less than $2\sqrt{r}$. Hence, the algorithm achieves welfare which is less than $2\sqrt{2}/\sqrt{n}$ of the optimum welfare, which can be made arbitrarily small by taking large enough number of items n. □