Recruitment Strategies That Take a Chance

Gregory Kehne
Harvard University

Ariel D. Procaccia
Harvard University

Jingyan Wang
Georgia Institute of Technology

Abstract

In academic recruitment settings, including faculty hiring and PhD admissions, committees aim to maximize the overall quality of recruited candidates, but there is uncertainty about whether a candidate would accept an offer if given one. Previous work has considered algorithms that make offers sequentially and are subject to a hard budget constraint. We argue that these modeling choices may be inconsistent with the practice of academic recruitment. Instead, we restrict ourselves to a single batch of offers, and we treat the target number of positions as a soft constraint, so we risk overshooting or undershooting the target. Specifically, our objective is to select a subset of candidates that maximizes the overall expected value associated with candidates who accept, minus an expected penalty for deviating from the target. We first analyze the guarantees provided by natural greedy heuristics, showing their desirable properties despite the simplicity. Depending on the structure of the penalty function, we further develop algorithms that provide fully polynomial-time approximation schemes and constant-factor approximations to this objective. Empirical evaluation of our algorithms corroborates these theoretical results.

1 Introduction

Anyone who has served on a faculty hiring committee or a PhD admissions committee knows that a successful outcome requires resolving the tension between two competing goals. On the one hand, some candidates are (perceived to be) better qualified than others, and the aim is to recruit the best candidates. On the other hand, there are a given number of positions to be filled, and while there is typically some flexibility, there is a real cost to recruiting too many or too few people. The tension arises in part because the stronger a candidate is, the more likely they are to receive multiple attractive offers and the less likely they are to accept any particular offer. In order to manage uncertainty, a good strategy may involve a mix of offers to stellar candidates and "safer" candidates.

To formalize this problem, we assume that a recruiting entity (academic or otherwise) has access to two numbers for each candidate i: their value x_i and their probability p_i of accepting an offer. We acknowledge that in current practice, these numbers are not always explicitly estimated. However, committees typically rank or assign numerical scores to candidates based on their strength or fit, and savvy committees roughly estimate recruitment chances by classifying candidates as, say, "high yield," "low yield" or "extremely low yield", for example, by past experience or assistive computational tools [12, 1]. Therefore, we believe that the gap between current practice and explicit value and probability estimates is not large.

Our approach builds on the work of Purohit et al. [11], who cast hiring under uncertainty as a stochastic optimization problem. In their basic model, there are n candidates (each associated with a value and probability), k positions, and t time steps. In each time step, the algorithm (i.e. recruitment strategy) may make an offer to a single candidate and receive a response; that is, at most t sequential offers can be made, and the budget of k cannot be exceeded. The goal is to maximize the expected value of candidates who accept offers. Purohit et al. also consider the setting where the algorithm may make parallel offers in each round. For both problems, they develop polynomial-time, constant-factor approximation algorithms (with approximations ratios of 2 and 8, respectively).

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

This problem formulation captures key aspects of recruitment, but, in our view, it does have two shortcomings. First, in a sense it is overcomplicated, as computational challenges stem from the assumption that offers are made sequentially: even with parallel offers, if there is a single time step $(t = 1)$ then it is optimal to make offers to the k candidates with largest $p_i \cdot x_i$. But, in our experience of faculty hiring and PhD admissions in several universities, offers are typically made in one batch. Indeed, delayed offers (in the case of faculty hiring) and waitlists (in the case of PhD admissions) are usually avoided as they negatively impact yield.

The second, and more crucial, shortcoming is that Purohit et al. [11] consider the constraint of hiring k candidates as firm. Again, this is inconsistent with our experience: Offers are made so that the expected yield roughly matches a desired target, but some faculty hiring or PhD admission cycles are "too successful," in the sense that the number of candidates who accept their offers is much larger than expected. This has a real cost: in the case of PhD students, it creates difficulties in finding funding and advisors, and in the case of faculty hiring, it may precipitate a shortage of resources with long-term impacts on future hiring and even tenure. For example, in one of our institutions, a faculty hiring cycle with yield that was much higher than expected led to the cancellation of the subsequent year's search.

Let us, therefore, reformulate the problem of hiring under uncertainty in a way that avoids both issues. We assume that offers are made in a single batch, and the target number of positions k is a soft constraint. Specifically, a penalty is incurred for deviating from the target number of positions; we consider several different options for this penalty function. The optimization problem is this:

> *Select a subset S of candidates that maximizes overall expected reward $\sum_{i \in S} p_i \cdot x_i$, minus expected penalty for deviating from the target number of positions.*

Enumerating all possible subsets S may be practicable for small instances, for example in the case of faculty hiring in small departments. However, a brute-force approach will not work for this purpose in larger departments, or at the scale of PhD admissions even in smaller programs, which motivates our search for good algorithms.

Our results. We first consider a simplified case where the goal is to solely minimize the penalty term of our objective (irrespective of the rewards), and show that the greedy algorithm that selects candidates in decreasing order of their probabilities is optimal (Section 3.1).

The full objective is considerably more complex, and we analyze it under two natural penalty functions. When the penalty function is the mean squared error from the target, we show that the optimization problem is weakly NP-hard, and provide a fully polynomial-time approximation scheme (FPTAS). When the penalty is linear in the extent to which the target is exceeded (that is, a linear penalty is incurred by overshooting, but not by undershooting), we show that two greedy heuristics — picking in the decreasing order of the value x_i and the expected value $p_i x_i$ — provide approximations to the optimal solution that are polynomial in the minimum probability $p_{\min}$. We then present a constant-factor approximation algorithm that runs in polynomial time for fixed $p_{\min}$ and candidate value relative to overshooting penalty, thereby improving upon the greedy heuristics.

Finally, we carry out experiments on synthetically generated data (Section 4), focusing on L_1^+ loss. We observe that the two greedy heuristics perform reasonably well, especially if the values and probabilities are positively correlated. At the same time, compared to the greedy heuristics, our constant-factor approximation algorithm better adapts to specific instances, especially when this correlation is negative. These numerical experiments corroborate our theoretical results that the greedy heuristics provide reasonable guarantees, justifying their use in practice for both simplicity and good performance. However in many regimes the additional flexibility of the constant-factor approximation algorithm is likely worth its complexity overhead.

Related work. In practice, the challenge of uncertainty in admissions is mitigated by practices such as admitting students in multiple rounds, using a waitlist, and using a rolling process [9]. There are also assistive computational tools that predict student yield rate with machine learning [12, 1]. As previously mentioned, there are a few theoretical formulations that model and address the uncertainty in such problems. Purohit et al. [11] consider an online setting where in each timestep if a candidate is given an offer then their decision is revealed immediately, and analyzes the optimal ordering to give the candidates offers to subject to a hard constraint on the total number of acceptances. Ganguly

et al. [7] consider a setting with multiple rounds where the yield rate in each round is either q_L or q_H (with $q_L < q_H$). To model the negative correlation between the candidate quality and the probability of acceptance, they assume that the probability of the yield rate being q_H is linear in the number of students admitted. They subsequently derive a decision tree that computes the number of admissions to make in each round. For single batch selection, Zhang and Pippins [13] analyze the optimal number of applicants to admit using techniques from yield management, under the assumption that each applicant has identical value and probability. A distinct line of work casts the admissions problem as a decentralized matching market [4, 5], where the uncertainty in acceptance is modeled by the students' stochastic preferences over multiple schools. An objective combining the utility and the accepted size is considered, but the penalty is on the expected size, not counting the variance involved as opposed to ours.

Our proposed formulation is closely related to the knapsack problem and its myriad variants. In the stochastic knapsack problem, each item has a deterministic value and an independent stochastic size; the actual size of an item is revealed only after it is selected [8, 6, 2]. For the one-shot version of this problem, the aim is to choose a subset maximizing the expected value of the realized items, such that the probability that these realizations violate the knapsack constraint is below some threshold. In contrast, our "items" have equal size, but we pay a penalty which is some function of our realized distance from our knapsack target size, exchanging the constraint for a mixed-sign objective. In this spirit, the one-sided loss functions we consider are similar to objectives which arise in the penalty method for solving constrained optimization problems [10], though our setting is stochastic and we do not introduce the penalty term in service of ultimately satisfying a hard constraint.

2 Problem Formulation

Taking a knapsack perspective, consider some n items (corresponding to candidates) with associated values $x_1, \dots, x_n \in \mathbb{R}$. If we select an item $i \in [n]$, there is a probability $p_i \in [0, 1]$ that we receive this item (the candidate accepts the offer). We write these values and probabilities as vectors $x \in \mathbb{R}^n$ and $p_i \in [0, 1]^n$. Let $Z_i \in \{0, 1\}$ be the indicator that we receive item i if it is selected, so that $Z_i \sim \text{Ber}(p_i)$. We assume the events that we receive each individual item are independent. Let $S_Z \subseteq S$ denote the random realization of chosen items; that is, $S_Z \coloneqq \{i \in S : Z_i = 1\}$. Our goal is to select a subset $S \subseteq [n]$. First, we consider the reward for a subset S as the expected total value obtained:

$$R(S) \coloneqq \mathbb{E}\left[\sum_{i \in S} Z_i x_i\right] = \sum_{i \in S} p_i x_i.$$

At the same time, let $M \in \mathbb{N}_+$ denote a target size that we want S_Z to achieve. We want to control the expected deviation of the realized size of S_Z, which is $|S_Z| = \sum_{i \in S} Z_i$, from the target size M. We consider this penalty as

$$V(S) \coloneqq \mathbb{E}\left[\rho\left(|S_Z|,\ M\right)\right],$$

where $\rho\colon \mathbb{N} \times \mathbb{N}_+ \to \mathbb{R}_{\geq 0}$ is a loss function, to be specified later. Combining the two parts, we define the overall objective as

$$U(S) \coloneqq R(S) - \lambda \cdot V(S), \tag{1}$$

where $\lambda \in \mathbb{R}_+$ is a hyperparameter that governs the importance of the penalty relative to the reward. Our goal is to find the subset that maximizes the overall expected utility:

$$S^* \in \operatorname*{arg\,max}_{S \subseteq [n]} U(S).$$

We denote a problem instance by $\mathcal{I} \coloneqq (x, p, M, \lambda)$, and the solution S^* is thus a function of the problem instance and the loss function ρ. It is worth noting that since the overall utility U is a mixed-sign objective, the optimal value of $U(S^*)$ may be negative depending on the instance and choice of ρ.

We consider a range of choices for the loss function ρ. Given a target size M, it is natural for ρ to be a convex function minimized at M, which penalizes any deviation from M, or alternatively a monotone convex function that is nonzero above M, which can be seen as penalizing violation of the budget constraint. We focus in particular on one- and two-sided linear and quadratic losses, which are formally introduced in Section 3.2 below.

3 Theoretical Results

To begin we note that if we only consider the reward term and set the penalty term to be $V(S) \coloneqq 0$, then the solution is to trivially select all items. In what follows, we first discuss the other extremal case, taking $R(S) \coloneqq 0$ and considering the penalty V in isolation. These may be viewed as the extreme cases when $\lambda = 0$ and $\lambda \to \infty$. We will then turn to the general objective and consider both terms jointly.

3.1 Warm-Up: Penalty Only

To gain intuition for this problem, we start with the simplified case in which our goal is only to minimize the penalty term. Note that in this case our objective is strictly nonpositive.

Algorithm 1 PGREEDY

Require: $p \in [0, 1]^n$
$S \leftarrow \emptyset$.
Sort $i \in [n]$ in decreasing order and re-index the items such that $p_1 \geq \ldots \geq p_n$
for $i = 1, 2, \ldots, n$ **do**
  if $U(S \cup \{i\}) \geq U(S)$ **then**
    $S \leftarrow S \cup \{i\}$
  else
    break
return S

We consider PGREEDY, the greedy algorithm with respect to p_i (Algorithm 1). In words, PGREEDY selects items in their decreasing order of probabilities, with ties broken arbitrarily if there are multiple items with the same probability.[1] The algorithm keeps selecting the next item defined by this order, and terminates when adding the next item would decrease the objective. As usual this greedy algorithm is computationally efficient, since the stopping criterion can be checked in polynomial time given access to ρ. (See Lemma 5 in Appendix B.8 for details.) Surprisingly, PGREEDY is optimal for minimizing $V(S)$ in isolation.

Proposition 1. *Let $M \in \mathbb{N}_+$ be any target size. If the loss function $\rho(\cdot, M)$ is convex, then* PGREEDY *(Algorithm 1) yields an optimal solution to minimizing the penalty $\min_{S \subseteq [n]} V(S)$.*

The proof of this proposition is provided in Appendix B.3. This result is not obvious, as one might expect that as the sum of probabilities of all selected items so far approaches the target size, it may be better to select an item with lower probability than an item with higher probability to "fill the gap." This is not true. Intuitively, it is because the realization of each Z_i is binary, so the outcome of adding another item i into the selection is either we add this item (with probability p_i) or not (with probability $1 - p_i$). If adding this item gives lower penalty, then we desire to add the item with the largest probability possible.

3.2 The General Objective

We now turn to the general objective. At the outset it bears noting that $U(S)$ is submodular in S whenever $\rho(\cdot, M)$ is convex (see Lemma 1 in Appendix B.1), as is the case for the loss functions we consider. Unfortunately the existing body of work on (non-monotone) submodular maximization cannot be leveraged to obtain a general-purpose approximation to $U(S)$, since U is mixed-sign and may be negative even at optimality, and applying an affine transformation in order to engineer nonnegativity will generally destroy any approximation guarantees.

We focus on a few natural choices of the loss function. First, we consider ρ given by linear and quadratic losses, which we denote by L_1 and L_2 respectively. These yield penalty terms $V(S)$ which are equal to the mean average error (MAE) and mean squared error (MSE) for the realized size of the

[1] In practice it is natural to break ties in favor of items of higher value, though this does not affect our results.

subset S_Z. We also consider the corresponding one-sided losses, defined by

$$L_1^+(|S_Z|, M) \coloneqq \begin{cases} |S_Z| - M & \text{if } |S_Z| \geq M \\ 0 & \text{otherwise,} \end{cases} \qquad \text{and} \qquad L_2^+(|S_Z|, M) \coloneqq L_1^+(|S_Z|, M)^2.$$

All of these losses considered penalize the case where the realized size is greater than the target size M. In applications such as admissions and hiring, there is a limited, pre-specified amount of resources allocated to the newly admitted or hired people. Hence, having more people than the target size is not desired. At the same time, the two-side losses give explicit preference that the realized size should also not be smaller than the target size. This explicit penalty for undershooting could represent a hit to morale (an unsuccessful recruitment cycle really is demoralizing) or insufficient staffing for required tasks, such as teaching certain courses. The one-side loss functions may to some extent capture these considerations, as there is an implicit opportunity cost described by the reward term when fewer candidates accept.

3.2.1 An FPTAS for L_2 Loss

Given that we understand our problem in both extremal cases (when only considering the reward term or the penalty term), one might hope that some interpolation between them could solve the general case. However the general case is more complicated. Recall that for the penalty-only objective, pGREEDY in Algorithm 1 attains the optimal selection, by adding items in decreasing order of p_i, and terminates once the next item strictly decreases the objective. But pGREEDY is clearly ill-suited to the general objective, since it does not take values into consideration. We now present two more natural greedy heuristics analogous to pGREEDY, and show that they are provably not optimal for the general objective. Specifically, we consider:

- xGREEDY: adds items in decreasing order of their value x_i, and terminates once the next item in this order strictly decreases the objective.
- xpGREEDY: adds items in decreasing order of their expected reward $x_i p_i$, and terminates once the next item in this order strictly decreases the objective.

Despite these heuristics appearing intuitive, they perform in a certain sense arbitrarily poorly even for the squared L_2 loss, as formalized by the following result.

Proposition 2. *Consider the two-sided loss $\rho = L_2$ and any $\lambda > 0$. Then for* pGREEDY, xGREEDY *and* xpGREEDY, *there exists an instance such that the algorithm selects $S \subseteq [n]$ for which $U(S) \leq 0$, while $U(S^*) > 0$.*

The proof of this proposition is provided in Appendix B.4, and we now provide an informal description of the instances constructed. Since pGREEDY does not take into account the item values x_i at all, it is natural to expect that pGREEDY is not suitable for the general objective. Specifically, we consider two items where one item has probability 1 and value 0, and the other item has a "good" probability less than 1 and a "good" value. Then pGREEDY selects the first item, whereas selecting the second item only yields a positive objective value. For xGREEDY and xpGREEDY, we consider two items that have almost the same expected reward. We let item 1 has a probability of 1. We let item 2 have a slightly greater expected reward for tie-breaking, and let item 2 have a smaller probability. In this case, xGREEDY and xpGREEDY start by picking item 2, which introduces nontrivial variance. When λ becomes large, this variance drives the overall objective negative. On the other hand, picking item 1 yields a strictly positive objective.

Despite the failure of heuristic approaches, when the chosen loss function is $\rho = L_2$ our problem can in fact be approximated up to negligible additive error. For this loss, our full objective (1) may be written as

$$U(S) = \sum_{i \in S} p_i x_i - \lambda \cdot \mathbb{E}\left(\sum_{i \in S} Z_i - M\right)^2. \tag{2}$$

Letting $b_i \in \{0, 1\}$ be the binary decision variable for whether $i \in S$, that is, $b_i \coloneqq \mathbb{1}\{i \in S\}$, the optimization problem then becomes

$$\arg\max_{S \in [n]} U(S) = \arg\max_{b \in \{0,1\}^n} \quad \sum_{i \in [n]} b_i p_i x_i - \lambda \cdot \mathbb{E}\left(\sum_{i \in [n]} b_i Z_i - M\right)^2. \tag{3}$$

Expanding (3) yields a collection of terms which are constant, linear, and quadratic in the b_i, and so the objective can be reformulated as an unconstrained binary quadratic program (UBQP). Although UBQP is strongly NP-hard, Çela et al. [3] present a pseudo-polynomial time algorithm for UBQP when the coefficient matrix for the quadratic form in the objective has constant rank. By proving that the objective (2) is sufficiently insensitive to small changes in our problem parameters, we leverage this pseudo-polynomial time algorithm to derive a FPTAS for approximating the optimal objective value for our problem. A standard search-to-decision reduction then yields the following result.

Theorem 1. *For $\rho = L_2$, Algorithm 4 identifies some $S \subseteq [n]$ satisfying $U(S) \geq U(S^*) - \epsilon$ in time $poly(1/\epsilon, n, M, \lambda)$.*

Pseudocode describing Algorithm 4 and the proof of this theorem are provided in Appendix B.5. On the other hand, by a reduction from equipartition we have the following hardness:

Theorem 2. *For $\rho = L_2$, optimizing $U(S)$ is weakly NP-hard.*

The proof of this theorem is provided in Appendix B.6. The hardness landscape of our problem when $\rho = L_2$ is therefore similar to that of the knapsack problem, which is heartening since the knapsack problem is relatively tractable in practice. However unlike the knapsack problem, we should only hope for additive rather than multiplicative guarantees; this is because for $\rho = L_2$ our optimal value is not bounded away from zero and may be strictly negative, even if all values x_i are nonnegative.

In contrast to L_2, we find that the one-sided loss L_2^+ is not straightforward to analyze. In this case the objective does not admit a quadratic factorization in terms of decision variables, and although the objective is nonnegative, it is difficult to analyze the performance of the greedy algorithm or contend with the nonlinearity of the loss function in a principled way. We instead turn to L_1^+ loss, where surprisingly these obstacles can be overcome.

3.2.2 Approximations for L_1^+ Loss

The loss $\rho = L_1^+$ enables the possibility of a multiplicative approximation because the optimum value $U(S^*)$ of the mixed-sign objective is nonnegative. More generally, any ρ with $\rho(0, M) = 0$ has a nonnegative optimal value, since in this case $U(\emptyset) = 0$. For $\rho = L_1^+$ choosing any single item i with positive x_i and p_i has strictly positive objective, since $M \geq 1$ implies that $L_1^+(1, M) = 0$ and so $U(\{i\}) = p_i x_i$. More generally, for this loss it suffices to consider only i for which $x_i > 0$, since under this loss the marginal contribution of any i with $x_i \leq 0$ is nonpositive.

The L_1^+ loss may appear amenable to a greedy algorithmic approach, since early items incur no penalty and the marginal penalty of adding a later item i is simply proportional to the probability that the current solution exceeds the target M. However, as in the case for the L_2 loss, these natural heuristics fail to consider the relation between items in the selection. While efficient, these greedy algorithms perform arbitrarily badly compared to the optimal solution in the worst case. The failure of PGREEDY is again apparent as in the case for the L_2 loss. We now provide some informal "bad" instances for XGREEDY and XPGREEDY for intuition.

Recall that XGREEDY chooses items in order by value. Consider $M = 1$, and consider two types of items with $(x_1, p_1) = (1, p)$ and $(x_2, p_2) = (0.5, 1)$. XGREEDY picks item 1 and yields an objective of $O(p)$, whereas picking item 2 yields a constant objective. The other greedy algorithm XPGREEDY chooses items i in the order of their expected value $x_i p_i$. We consider the instance with $M = 1$, and two types of items $(x_1, p_1) = (1 + \epsilon, 1)$ and $(x_2, p_2) = (1/p, p)$ with some tiny $\epsilon > 0$ so that XPGREEDY chooses an item from type 1 and yields a constant objective. On the other hand, choosing $\frac{1}{p}$ copies of item 2 yields an objective of $\Omega(1/p)$.

For both XGREEDY and XPGREEDY, the constructed instances yield an upper bound on the approximation ratio, scaling as p, where p is the smallest probability associated with the items. It suggests that the minimum probability $p_{\min} := \min_{i \in [n]} x_i$ is a natural parameter for measuring the complexity of an instance with respect to the L_1^+ loss. Another natural parameter is $x_{\max} := \max_{i \in [n]} x_i$, the maximum value among all items. Surprisingly, the performance of these greedy algorithms can be lower-bounded in terms of $p_{\min}$ as well, under an additional assumption about the values of items relative to λ.

Theorem 3. *Consider the one-sided loss $\rho = L_1^+$. If there is some fixed constant $c > 0$ such that $x_{max} \leq (1 - c) \cdot \lambda$, then*

(a) There exist instances for which PGREEDY *selects S with $U(S) = 0$, while $U(S^*) > 0$.*

(b) The worst-case approximation ratio for XPGREEDY *is $\Theta(p_{min})$.*

(c) The worst-case approximation ratio for XGREEDY *is $\Omega(p_{min}^2)$ and $O(p_{min})$.*

The proof of this theorem is provided in Appendix B.7. As a consequence, the approximation ratios of these greedy algorithms can be arbitrarily small as $p_{\min} \to 0$. Also note that these upper bounds do not require the assumption that $x_i \leq (1-c)\lambda$; in all three cases there exist bad instances irrespective of this assumption. The reason we can derive better lower bounds for XPGREEDY than XGREEDY is intuitive; this is because XPGREEDY measures the expected reward conferred by an item, and so can be more directly related to the optimal solution S^*.

This is in notable contrast to when $\rho = L_2$, where we saw that multiplicative guarantees are inapt and all three greedy algorithms may incur arbitrarily large additive loss. Theorem 3 also raises a natural question: is this $1/p_{\min}$ threshold tight, or is it possible to efficiently attain better approximations to $U(S^*)$ which do not depend on $p_{\min}$? We address this by introducing ONESIDEDL$_1^+$, presented in Algorithm 2, which attains a constant-factor approximation to the optimal solution. In this pseudocode, for any vector $v \in \mathbb{R}^n$ and set $S \subseteq [n]$, we use $v|_S$ to denote the $|S|$-dimensional vector obtained by restricting v to its coordinates indexed by S. Its runtime is parameterized by the minimum probability $p_{\min}$ and the ratio of the maximum value $x_{\max}$ to the penalty parameter λ; in particular for fixed $p_{\min}$ and $\lambda/x_{\max}$ it is polynomial in n.

At a high level, ONESIDEDL$_1^+$ proceeds first by dividing the items into three groups according to their values x_i, and considering each group in turn. Since U is submodular (Lemma 1), the optimal solution within at least one of these groups is constant-competitive with $U(S^*)$. We obtain a constant-factor approximation for each group in a different way.

For the items with low values bounded away from λ, LOWVALUEL$_1^+$ (Algorithm 5 in Appendix B.8) checks all small subsets, which succeeds if the optimal subset for this group is small. It also computes rounded probabilities and values (q_i, y_i) for each item in this group, and then efficiently computes the optimal solution according to this rounded instance. If the optimal subset is large, we then prove that this search over rounded solutions necessarily identifies a subset with objective value comparable to that of the optimal subset. This is the technical crux of proving that ONESIDEDL$_1$ is a constant-factor approximation.

For the items with values just below λ, MEDIUMVALUEL$_1^+$ (Algorithm 6 in Appendix B.8) returns the optimal subset if the group is small. If the group is large, it tries to choose a subset such that the expected number of realizations is about M; if there are not enough it chooses a subset with approximately half the expected realizations of the group overall. Finally for the group of items with values above λ, it is easy to see that choosing the entire group is optimal. The pseudocode and related proofs for these algorithms appear in Appendix B.8. The following result provides a theoretical guarantee for Algorithm 2.

Algorithm 2 ONESIDEDL$_1^+$

Require: Problem instance $\mathcal{I} = (x, p, M, \lambda)$
Ensure: $S \subseteq [n]$ for which $U(S) \geq c \cdot U(S^*)$ for universal constant c
1: $N_L \leftarrow \{i \in [n] \colon x_i \leq (1 - \frac{p_{\min}}{4}) \cdot \lambda\}$
2: $N_M \leftarrow \{i \in [n] \colon (1 - \frac{p_{\min}}{4}) \cdot \lambda < x_i < \lambda\}$
3: $N_H \leftarrow \{i \in [n] \colon x_i \geq \lambda\}$
4: $S_L \leftarrow$ LOWVALUEL$_1^+(x|_{N_L}, p|_{N_L}, \lambda, M)$
5: $S_M \leftarrow$ MEDIUMVALUEL$_1^+(x|_{N_M}, p|_{N_M}, \lambda, M)$
6: $S_H \leftarrow N_H$
7: Compute $U(S_L)$, $U(S_M)$, and $U(S_H)$
8: **return** $S \in \{S_L, S_M, S_H\}$ maximizing $U(S)$

Theorem 4 (Constant-factor approximation for L_1^+)**.** *Algorithm 2 is a constant-factor approximation to $U(S^*)$ which runs in time $n^{O\left(\frac{1}{p_{min}^2} \max\left\{1, \log\left(\frac{1}{p_{min}}\right), \log\left(\frac{\lambda}{x_{max}}\right)\right\}\right)}$.*

The proof of this theorem is provided in Appendix B.8. Intuitively, the reason Algorithm 2 divides the items into cases depending on their values is to handle the case when the reward portion of $U(S^*)$ is almost equal to the penalty portion. This presents an impediment to the efficient strategy of solving a rounded version of the instance, since in this case the magnitude and even the sign of $U(S^*)$ is potentially quite sensitive to changes in the p_i and x_i. By restricting attention to the low-value case, we prove that the expected number of realized items in S^* is not much more than the threshold M. This then allows us to argue that there exist good rounded solutions, which we may efficiently identify.

We conclude our theoretical results with a surprising equivalence between one- and two-sided linear losses L_1^+ and L_1. In what follows, we use U_{L_1} and $U_{L_1^+}$ to denote the objective with $\rho = L_1$ and $\rho = L_1^+$, respectively. We use $U(S;\, \mathcal{I})$ to denote the evaluation of $U(S)$ specifically with respect to the instance $\mathcal{I} = (x, p, \lambda, M)$.

Theorem 5 (Equivalence between L_1 and L_1^+)**.** *For any instance $\mathcal{I} = (x, p, \lambda, M)$ and for all $S \subseteq [n]$, consider $\mathcal{I}' = (x', p, \lambda', M)$ given by $x_i' := x_i - \lambda$ and $\lambda' := 2\lambda$. Then*

$$U_{L_1}(S;\, \mathcal{I}) = U_{L_1^+}(S;\, \mathcal{I}') - \lambda \cdot M.$$

The proof of this theorem is provided in Appendix B.9. In particular, since λ and M do not depend on S, this implies that S maximizes U_{L_1} on $\mathcal{I}$ if and only if it maximizes $U_{L_1^+}$ on $\mathcal{I}'$.

Although Theorem 5 establishes a correspondence between the solutions to our problem for $\rho = L_1$ and $\rho = L_1^+$, it is not approximation preserving and so does not convert OneSidedL$_1$ into an approximation algorithm for the two-sided setting. Indeed, as in the $\rho = L_2$ setting, the optimal value when $\rho = L_1$ can be strictly negative.

4 Numerical Experiments

Having established worst-case theoretical guarantees, we wish to test how well our algorithms perform empirically. We focus on L_1^+ loss because our result for L_2 is an FPTAS, so we know its performance can be made to be arbitrarily close to optimal. Specifically, the experiments benchmark the subroutine LowValueL$_1^+$ (part of OneSidedL$_1^+$) against xGreedy and xpGreedy for the regime where $x_i \leq (1-c)\lambda$. This is the regime which LowValueL$_1^+$ was developed to handle for OneSidedL$_1^+$, and it is the regime for which we prove performance guarantees for xpGreedy and xGreedy. All error bars shown in the plots represent standard error of the mean. The code to reproduce our simulationresults is available at `https://github.com/jingyanw/recruitment_uncertainty`.

4.1 Experimental Setting

In constructing instances we follow the approach of Purohit et al. [11] in their use of beta distributions to orchestrate different kinds of correlation between x_i and p_i. We therefore first draw $x_i \sim \text{Unif}[0, 1]$, and then produce three types of correlation as follows:

- *Negative correlation:* $p_i \sim p_{\min} + (1 - p_{\min}) \cdot \text{Beta}(10(1 - x_i),\ 10x_i)$.
- *Positive correlation:* $p_i \sim p_{\min} + (1 - p_{\min}) \cdot \text{Beta}(10x_i,\ 10(1 - x_i))$.
- *No correlation*: $p_i \sim \text{Unif}[p_{\min}, 1]$.

This construction differs from the sampling paradigm of Purohit et al. [11] only in that we re-normalize the probabilities $\{p_i\}$ so that they are bounded in $[p_{\min}, 1]$. We consider $n = 50$ and $p_{\min} = 0.01$ throughout, and explore the greedy heuristics xGreedy and xpGreedy, as well as the constant-factor approximation algorithm OneSidedL$_1^+$ (Algorithm 2), for a range of M and λ. This lower bound on $p_{\min}$ ensures that the performance of the greedy heuristics and runtime of our algorithm are reasonable; a value of 0.01 (say) is realistic because in practice, if a candidate takes the time and effort to apply, it is reasonable to assume that they at least have some nontrivial probability to accept if they were given an offer. We also focus on the regime where $x_i < \lambda$, which is assumed by Theorem 3 and handled in Algorithm 2 by the subroutine LowValueL$_1^+$. We believe that this is the main regime of practical interest: candidates with $x_i \geq \lambda$ are beneficial regardless of how many candidates have already accepted offers, and one might suppose that such candidates are rare.

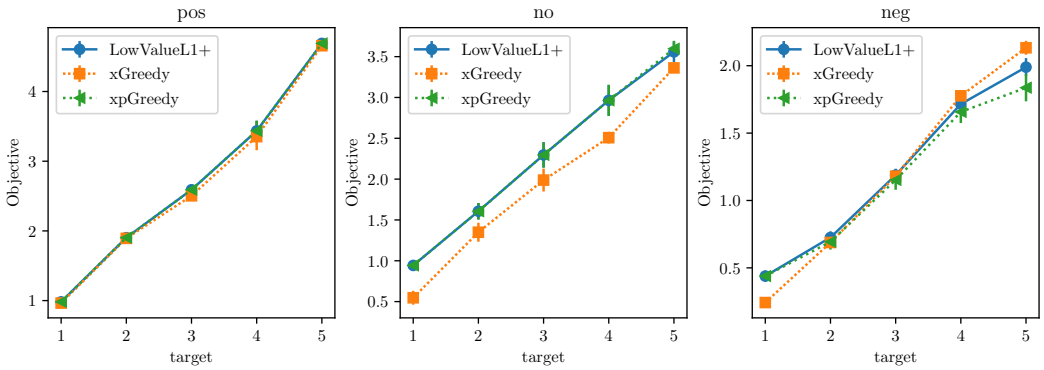

Figure 1: Sampling from the beta distribution with positive, no, and negative correlation. Here $n = 50$ and $\lambda = 3$.

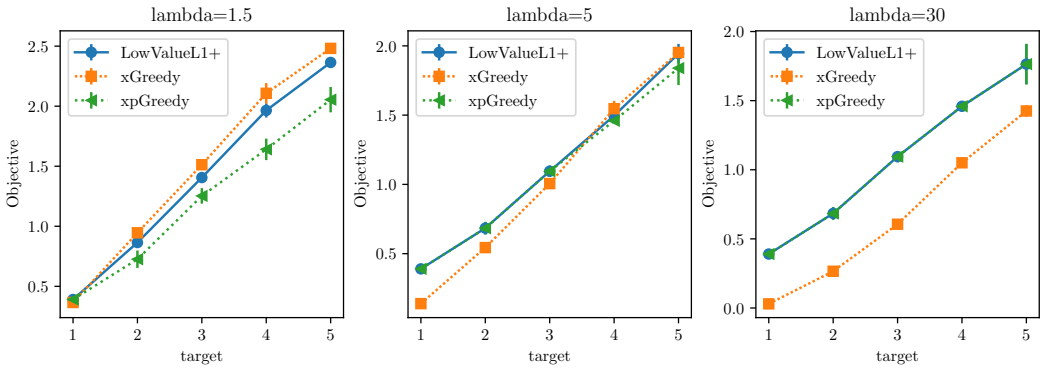

Figure 2: Performance for increasing penalty regularizer λ. Here $n = 50$ and sampling is via the negatively correlating beta distribution.

Note that our theoretical guarantees (Theorem 4) necessitate that all candidate solutions up to size $\tau = \widetilde{O}(1/p_{\min}^2)$ are checked by brute force. In this implementation of LowValueL$_1^+$ we take $\tau = 0$ and isolate its search over rounded solutions. As we consider small target sizes M, this prevents LowValueL$_1^+$ from outperforming the greedy algorithms simply by virtue of having brute forced over all relevant solutions. This only hinders the performance of LowValueL$_1^+$. In fact, LowValueL$_1^+$ only considers rounded solutions S satisfying $\sum_{i \in S} x_i p_i \leq 2M$. So long as $x_i < \lambda$, such a stopping condition can be deployed without loss of generality. For details, see Appendix B.8. We believe this offers a favorable tradeoff between runtime and accuracy, and illustrates a lower bound on the performance of LowValueL$_1^+$ as written.

4.2 Experimental Results

The objective values that our algorithms of interest attain for these distributions are shown in Figure 1. Note that positive correlation leads xGreedy and xpGreedy to pursue very similar (and optimal) strategies, as expected. This is intuitively the easier setting, and here LowValueL$_1^+$ performs on par with the greedy heuristics. In the no-correlation and negative-correlation settings, there are regimes where one of the two greedy heuristics performs better than the other one, whereas LowValueL$_1^+$ appears to perform as well as the better of two depending on the regimes, showing its better adaptivity across these instances in practice as well as in theory.

Negative correlation between x_i and p_i is of particular interest to us, since it seems most relevant for the setting of faculty hiring and PhD admissions, and in fact hiring and recruitment more broadly. In Figure 1, we also observe that negative correlation is the setting that displays the most heterogeneity in algorithm behavior. We therefore turn to this negative-correlation setting and explore the effect of increasing the penalty regularizer λ in Figure 2. In general, LowValueL$_1^+$ appears comparable to

the better of the two greedy approaches across the values of M and λ that we examine, though there is a small gap between the objectives achieved by LOWVALUEL_1^+ and XGREEDY when $\lambda = 1.5$.

This is also good news for XGREEDY and XPGREEDY, because it suggests that the two of them together remain competitive across a wide range of instances. To the extent that LOWVALUEL_1^+ falls short of $U(S)$, it is due to systematically rounding the probabilities p_i up by a constant factor when computing the prospective utility of solutions. Because its rounding preserves the reward term, such a systematic overestimate in the p_i leads it to overestimate the penalty term of any set under consideration. The presence of a large number of (x_i, p_i) with individually balanced but collectively large impact on the objective could therefore explain the extent to which LOWVALUEL_1^+ lags behind XGREEDY in Figure 2; the latter chooses many such individuals while the former judges their influence on the penalty to be too large. However, this can be mitigated by choosing smaller multiplicative bucket sizes for LOWVALUEL_1^+, which is particularly effective in the case where the $\{p_i\}$ of an instance fall in a small number of clusters or exhibit other structure.

5 Discussion

One of the takeaways from our theoretical and empirical results is that the greedy algorithm XGREEDY, which makes offers to a subset of candidates with the highest values, is practicable for L_1^+ loss. This is intriguing because the algorithm is quite similar to how faculty hiring and admissions committees typically think: they want to make offers to the best candidates. The difference is that XGREEDY carefully selects the *number* of offers to be made, in a way that (greedily) maximizes the objective. Since XGREEDY amounts to a relatively small tweak to current practice, we believe committees would find the algorithm to be especially palatable.

An issue our results do not address is which penalty function best matches the needs of a specific recruitment process. For example, is there a rigorous way to argue that a particular choice of penalty function is more broadly applicable than another? That said, the choice between one-sided and two-sided penalty is rather intuitive, depending on the application. And our results provide computational arguments in favor of L_2 when two-sided penalty is desired, and L_1^+ for one-sided penalty.

From an ethical viewpoint, a potential concern is that our proposal may ultimately have unintended negative consequences. For example, if many faculty hiring committees adopted our optimization-based approach, might candidates have fewer opportunities? We believe, however, that the opposite is true. Currently the academic job market is strikingly inefficient, as committees often converge on a few candidates who are inundated with interviews and offers, while comparably strong candidates are left with nothing. If our approach is adopted widely, it is likely to widen the pool of candidates who receive appealing offers. Granted, a centralized matching market (in the style of the National Resident Matching Program) may be an even better solution, but creating such a market requires a huge — and often impractical — degree of coordination; by contrast, our approach can be adopted independently by institutions and even by individual departments or committees.

Acknowledgments. We thank Anupam Gupta and Alejandro Toriello for helpful discussions. GK and AP were partially supported by the National Science Foundation under grants IIS-2147187, CCF-2007080, IIS-2024287, and CCF-1733556; and by the Office of Naval Research under grant N00014-20-1-2488. JW was supported by the Ronald J. and Carol T. Beerman President's Postdoctoral Fellowship and the ARC (Algorithms & Randomness Center) Postdoctoral Fellowship at Georgia Tech.

References

[1] John Abelt, Daniel Browning, Celia Dyer, Michael Haines, Jalen Ross, Patrick Still, and Matthew Gerber. Predicting likelihood of enrollment among applicants to the UVa undergraduate program. In *Systems and Information Engineering Design Symposium*, pages 194–199, 2015.

[2] Anand Bhalgat, Ashish Goel, and Sanjeev Khanna. Improved approximation results for stochastic knapsack problems. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1647–1665, 2011.

[3] Eranda Çela, Bettina Klinz, and Christophe Meyer. Polynomially solvable cases of the constant rank unconstrained quadratic 0-1 programming problem. *J. Comb. Optim.*, 12(3):187–215, 2006.

[4] Yeon-Koo Che and Youngwoo Koh. Decentralized college admissions. *Journal of Political Economy*, 124(5):1295 – 1338, 2016.

[5] Xiaowu Dai and Michael I. Jordan. Learning strategies in decentralized matching markets under uncertain preferences. *Journal of Machine Learning Research*, 22(260):1–50, 2021.

[6] B.C. Dean, M.X. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 208–217, 2004.

[7] Subhamoy Ganguly, Ranojoy Basu, and Viswanathan Nagarajan. A binomial decision tree to manage yield-uncertainty in multi-round academic admissions processes. *Naval Research Logistics (NRL)*, 69(2):303–319, 2022.

[8] Jon Kleinberg, Yuval Rabani, and Éva Tardos. Allocating bandwidth for bursty connections. *SIAM Journal on Computing*, 30:191–217, January 2000.

[9] Yao Luo and Yu Wang. Dynamic decision making under rolling admissions: Evidence from U.S. law school applications. Technical Report tecipa-681, University of Toronto, Department of Economics, 2020.

[10] Anne L. Olsen. Penalty functions and the knapsack problem. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, June 27-29, 1994*, pages 554–558. IEEE, 1994.

[11] Manish Purohit, Sreenivas Gollapudi, and Manish Raghavan. Hiring under uncertainty. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 5181–5189, 2019.

[12] Linda Siefert and Fred Galloway. A new look at solving the undergraduate yield problem: The importance of estimating individual price sensitivities. *College and University*, 81(3):11–17, 2006.

[13] Li Zhang and John C. Pippins. Application of overbooking model to college admission at the citadel. *International Transactions in Operational Research*, 28(5):2402–2413, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Additional Experiments

In this section, we present additional experiments which shed more light on the performance of xGreedy, xpGreedy, and OneSidedL$_1^+$ relative to one another and to the optimal solution, for a broader range of objectives. The family of distributions from which we sample instances is the same as the one described in Section 4.1.

A.1 Comparison to Optimal

First, we recreate Figure 1 and Figure 2, now including the objective value of the optimal solution S^* as a benchmark for the three algorithms considered above. Since determining $U(S^*)$ by brute force is computationally costly, this comparison is undertaken for smaller instances ($n = 20$). Following Section 4.1, we consider $\lambda = 3$.

Here Figure 3 shows the performance of xGreedy, xpGreedy, and OneSidedL$_1^+$ relative to the objective $U(S^*)$ of the optimal solution, when values and probabilities are positively correlated, uncorrelated, and negatively correlated, for a range of target sizes M. Figure 4 shows the performance of xGreedy, xpGreedy, and OneSidedL$_1^+$ relative to $U(S^*)$ as the penalty regularizer increases, for negatively correlated x_i and p_i and again for a range of target sizes M.

It is noteworthy that in both Figure 3 and Figure 4, the best algorithms in each setting nearly attain the optimal objective value. It is unclear the extent to which we should expect that this continues to hold for larger instances, where solving the optimal solution by brute force is computationally infeasible.

A.2 Other Objectives

In Section 4.2 and Section A.1, we examine the performance of different algorithms for the L_1^+ loss, since this is the loss function for which we derive worst-case multiplicative guarantees and for which the algorithm OneSidedL$_1^+$ was designed.

However, it is still interesting to investigate how these algorithms perform with respect to other loss functions, despite the absence of worst-case theoretical guarantees for these greedy heuristics. Figure 5 compares the greedy heuristics between the L_1^+ and L_2^+ loss objectives across different correlation regimes. Figure 6 does the same for the two-sided L_1 and L_2 loss objectives.

In Figure 5 the performance of both greedy heuristics is very similar under the two one-sided losses. For the two-sided losses L_1 and L_2, Figure 6 suggests that the relative strengths of the two greedy heuristics remain roughly the same across the choice of two-sided loss. We observe that the objective values are no longer uniformly positive, and are no longer monotonically increasing in the target size. This is because the problem under the two-sided losses is fundamentally more difficult: Under one-sided losses, only selecting over the target is penalized; it is straightforward to observe that selecting M items always yields a penalty of 0 and hence a positive objective value. Under two-sided losses, selecting under the target and selecting over the target is both penalized; there is also non-

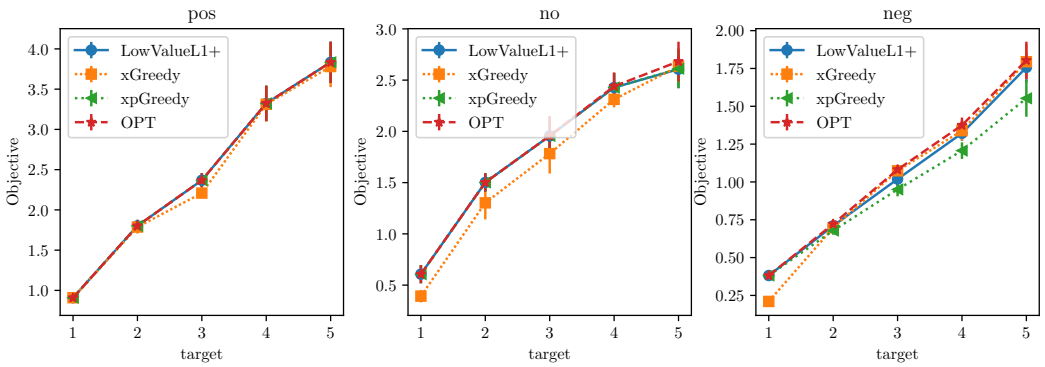

Figure 3: Sampling from the beta distribution with positive, no, and negative correlation. Here $n = 20$ and $\lambda = 3$, and OPT denotes $U(S^*)$.

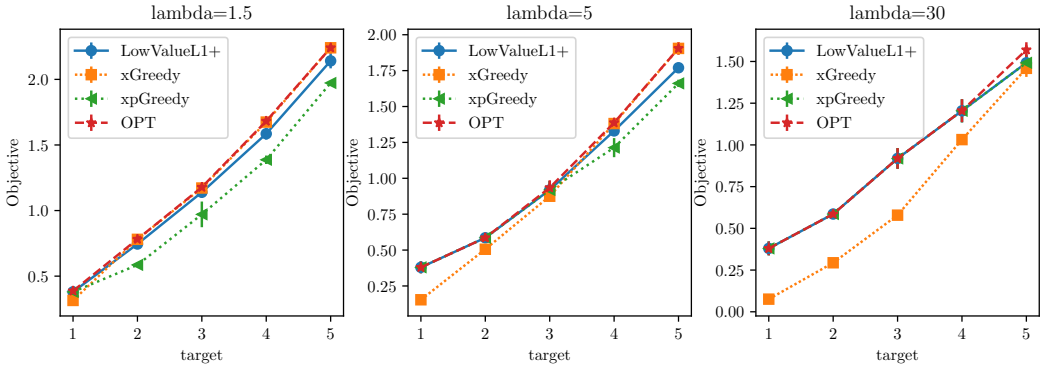

Figure 4: Performance for increasing penalty regularizer λ. Here $n = 20$ and sampling is via the negatively correlating beta distribution.

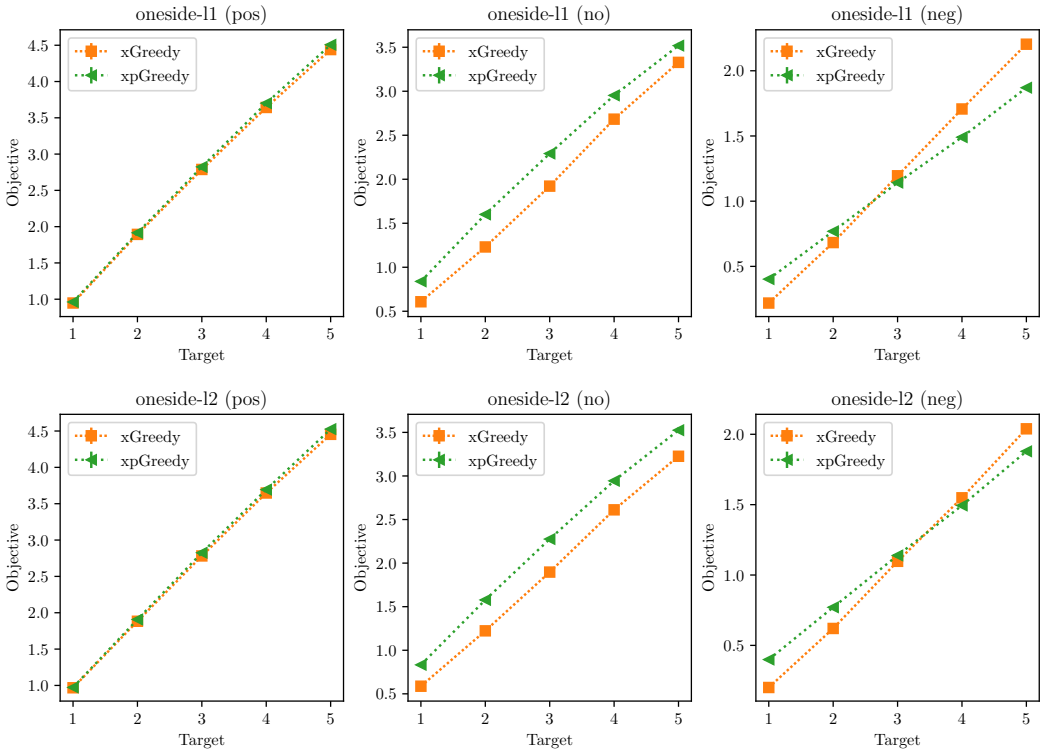

Figure 5: Evaluation of greedy heuristics for L_1^+ versus L_2^+ one-sided loss. Here $n = 50$ and $\lambda = 3$.

zero variance towards achieving the exact target M, and hence the objective is negative when the regularizer λ is large.

Comparing the two-sided losses L_1 and L_2, the problem under the L_2 loss is more difficult due to its higher penalty (the quadratic function always attains a higher value than the linear function on integers). The objective starts decreasing as a function of the target M: If we are aiming at a larger target M, more items are selected, leading to an inevitable increase in the variance and hence a lower objective.

We observe that XPGREEDY seems to dramatically outperform XGREEDY when the loss is two-sided. We provide an informal explanation, using the two-sided L_2 loss as an example. Under this loss, a candidate i contributes $x_i p_i$ to the reward term of the objective, while contributing $p_i(1 - p_i)$ to the variance of the realized size. When faced with two candidates of equal value x_i, we should therefore

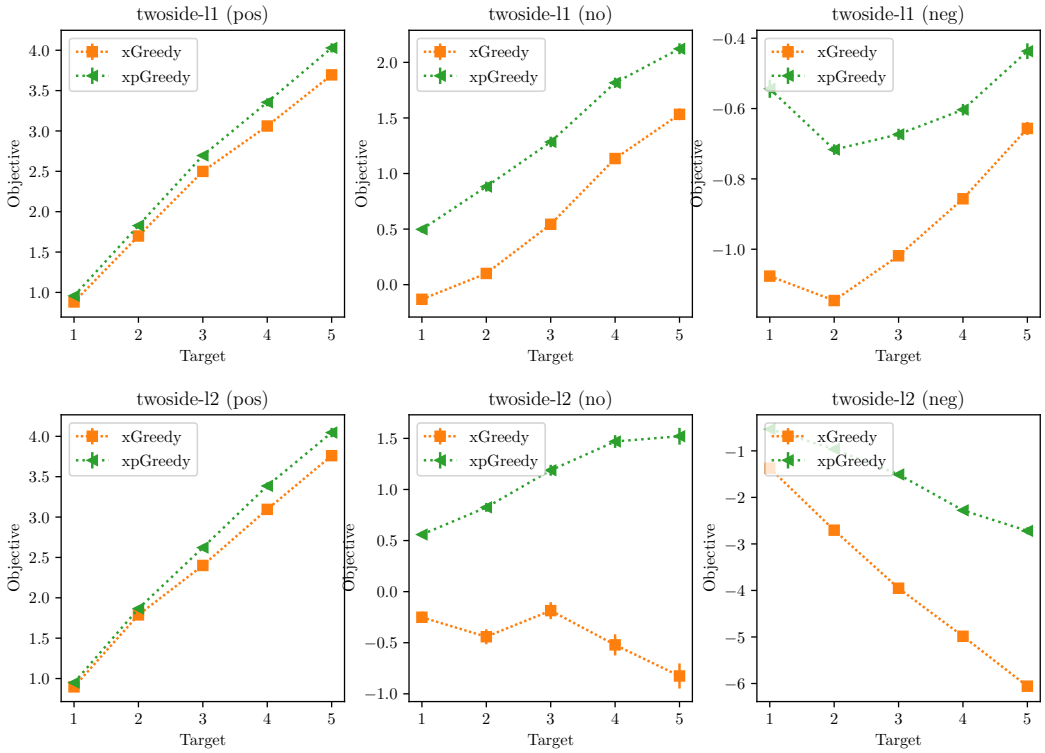

Figure 6: Evaluation of greedy heuristics for L_1 versus L_2 two-sided loss. Here $n = 50$ and $\lambda = 3$.

at the margin prefer the candidate with the higher probability, since this candidate contributes less to the variance per contribution to the reward. Note that for sufficiently large λ two-sided loss functions encourage algorithms to choose solutions expected size very close M, meaning that the variance and the two-sided L_2 loss are nearly equal. Here XPGREEDY prefers this higher-probability candidate, while XGREEDY is indifferent, explaining the superior performance of XPGREEDY.

B Proofs

In this section, we present the proofs of all theoretical results.

B.1 Preliminaries

For any set or event S, we use $\overline{S}$ to denote its complement. We use the notation $f(x) \lesssim g(x)$ to denote that there exists some universal positive constant $c > 0$, such that $f(x) \leq c \cdot g(x)$, and use the notation $f(x) \gtrsim g(x)$ when $g(x) \lesssim f(x)$.

For any vector $x \in \mathbb{R}^n$ and set $S \subseteq [n]$, we use the shorthand $x_S \coloneqq \{x_i\}_{i \in S}$. We say that S is a prefix of the ordered elements $\{a_1, \ldots, a_n\}$, if $S = \{1, 2, \ldots, k\}$ for some $k \in \{0, \ldots, n\}$. Let $\mu_S \coloneqq \mathbb{E}[\sum_{i \in S} Z_i] = \sum_{i \in S} p_i$. We also denote by $\mu^* \coloneqq \mu_{S^*}$ the expected size of the optimal subset.

The following lemma shows the submodularity of the objective U in the selection S.

Lemma 1. *If $\rho(\cdot, M)$ is convex then $U(S)$ is submodular in S.*

The proof of this lemma is provided in Appendix B.2. The submodularity is used for analyzing $\rho = L_1^+$ (in Section 3.2.2 and the proofs in Appendix B.8).

For the L_1^+ loss, the following lemma allows us to reason about the cardinality of S^* in the case when the penalty λ is larger than any of the values.

Lemma 2 (Mean Bound)**.** *There exists a universal constant $c_0 > 0$ such that the following is true. Consider the L_1^+ objective. For any $\epsilon \in (0, \frac{3}{4})$, if $x_{max} \le (1-\epsilon)\cdot\lambda$, then either*

$$|S^*| \le \frac{c_0 \log(\frac{1}{\epsilon})}{p_{min}} \tag{4a}$$

or

$$\mu^* \le \frac{101}{100} M. \tag{4b}$$

The proof of this lemma is provided in Appendix B.2.1. It is used in the proofs of Theorem 3 and Theorem 4. Intuitively, this lemma says that so long as all values are less than and bounded away from λ, either the optimal solution has expected size which is at most on the order of M, or the number of items in the optimal solution is small enough that concentration does not apply.

B.2 Proof of Lemma 1

We write out the objective $U(S)$ over all possible realizations of $Z \in \{0,1\}^n$:

$$\begin{aligned} U(S) &\coloneqq R(S) - \lambda \cdot \mathbb{E}\left[\rho\left(|S_Z|,\ M\right)\right] \\ &= \sum_{i\in S} p_i x_i - \lambda \sum_{z\in\{0,1\}^n} \mathbb{P}(Z=z)\cdot \rho\left(\sum_{i\in S} z_i,\ M\right). \end{aligned} \tag{5}$$

The first term in (5) is additive, and hence submodular. Since ρ is convex, it can be verified that the loss $\rho\left(\sum_{i\in S} z_i,\ M\right)$ is supermodular in S for each fixed realization z. Taking linear combinations of these terms yields the submodularity of $U(S)$. □

B.2.1 Proof of Lemma 2

Recall that S^* denotes the optimal solution under the L_1^+ objective. Denote by $i^* \coloneqq \arg\min_{i\in S^*} p_i$ the item in the optimal selection with the minimal probability, and denote $S \coloneqq S^* \setminus \{i^*\}$. In what follows, we prove claim (4) by deriving a lower bound and an upper bound on $\mathbb{P}(\sum_{i\in S} Z_i \le M)$.

Lower bounding $\mathbb{P}(\sum_{i\in S} Z_i \le M)$ by the optimality of S^*. Recall that the variance term penalizes the case when the total number of accepted items exceeds M. Hence, adding item i^* to the set S is only beneficial if $\mathbb{P}(\sum_{i\in S} Z_i \ge M)$ is small. Formally, we have

$$\begin{aligned} U(S^*) - U(S) &= x_{i^*}p_{i^*} - \lambda\mathbb{E}\left[(\sum_{i\in S} Z_i + Z_{i^*} - M)_+ - (\sum_{i\in S} Z_i - M)_+\right] \\ &= x_{i^*}p_{i^*} - \lambda p_{i^*}\cdot\mathbb{P}\Big(\sum_{i\in S} Z_i \ge M\Big) \end{aligned}$$

By the optimality of S^*, we have $U(S^*) \ge U(S)$, and hence

$$\begin{aligned} \mathbb{P}\Big(\sum_{i\in S} Z_i > M\Big) &< \mathbb{P}\Big(\sum_{i\in S} Z_i \ge M\Big) \\ &\le \frac{x_{i^*}p_{i^*}}{\lambda p_{\min}} \le \frac{x_{i^*}}{\lambda} \overset{\text{(i)}}{\le} 1-\epsilon, \end{aligned}$$

where step (i) is true by the assumption that $x_{\max} \le (1-\epsilon)\lambda$. Equivalently, we have

$$\mathbb{P}\Big(\sum_{i\in S} Z_i \le M\Big) \ge \epsilon. \tag{6}$$

Upper bounding $\mathbb{P}(\sum_{i\in S} Z_i \le M)$ by concentration. Let the universal constant c be such that $c \ge \frac{200}{\log(\frac{4}{3})}$. Then in (4a) we have $\frac{c\log(\frac{1}{\epsilon})}{p_{\min}} \ge 200$. Hence, it suffices to consider the case when $|S^*| \ge 200$. In what follows, we assume that condition (4b) does not hold. That is, we assume

$\mu^* > \frac{101}{100}M$. We prove that condition (4a) holds. To do so, we apply the multiplicative Chernoff bound on $\mathbb{P}(\sum_{i\in S} Z_i \le M)$. We first establish a relation between μ_S and M. Using the definition that i^* is the item with the smallest probability in the selection S^*, we have

$$\mu^* = \sum_{i\in S} p_i + p_{i^*} \le \frac{|S^*|}{|S^*|-1}\mu_S \overset{\text{(i)}}{\le} \frac{200}{199}\mu_S, \tag{7}$$

where step (i) uses the assumption that $|S^*| \ge 200$. Combining (7) with the assumption that $\mu^* > \frac{101}{100}M$, we have

$$\begin{aligned}\frac{101}{100}M < \mu^* &\le \frac{200}{199}\mu_S \\ M &< \frac{200}{199}\cdot\frac{100}{101}\mu_S \le (1-c_0)\mu_S,\end{aligned}$$

where $c_0 > 0$ is a universal constant. Since $\mu_S > M$, by the multiplicative Chernoff bound,

$$\mathbb{P}\Big(\sum_{i\in S} Z_i \le M\Big) \le \mathbb{P}\Big(\sum_{i\in S} Z_i \le (1-c_0)\,\mu_S\Big) \le \text{Exp}\left(-\frac{c_0^2\mu_S}{2}\right) \le \text{Exp}\left(-\frac{c_0^2\cdot|S|\cdot p_{\min}}{2}\right) \tag{8}$$

Now we combine the lower bound bound (6) and the upper bound (8) on $\mathbb{P}(\sum_{i\in S} Z_i \le M)$, we have

$$\epsilon \le \mathbb{P}\Big(\sum_{i\in S} Z_i \le M\Big) \le \text{Exp}\left(-\frac{c_0^2\cdot|S|\cdot p_{\min}}{2}\right)$$

and so

$$|S| \le \frac{2}{c_0^2}\cdot\frac{\log(\frac{1}{\epsilon})}{p_{\min}}.$$

By the assumption that $\epsilon \le 3/4$, we have $|S^*| = |S| + 1 \le \frac{c\log(\frac{1}{\epsilon})}{p_{\min}}$ for some universal constant $c > 0$, satisfying condition (4a). □

B.3 Proof of Proposition 1

We fix any target size $M > 0$. For notational simplicity, we use the shorthand $\rho(\cdot) \coloneqq \rho(\cdot, M)$ for the loss function. Recall that Algorithm 1 sorts the items in decreasing order of probability as $p_1 \ge \ldots \ge p_n$, with ties broken arbitrarily. In what follows, we first show that there exists a prefix of this ordering that is an optimal selection. Then we show that using the stopping criterion achieves the minimum variance among all prefixes, and hence is an optimal selection.

Step 1: A prefix in decreasing order of p_i achieves an optimal selection. Assume that there exists an optimal selection, denoted by $S^* \subseteq [n]$, that is not a prefix in decreasing order of p_i. By the assumption that S^* is not a prefix, there must exists items $i \in S^*$ and $i' \notin S^*$, such that $p_{i'} \ge p_i$. We now show that $S^{*\prime} \coloneqq S^* \cup \{i'\} \setminus \{i\}$, namely removing i from S^* and then adding i', also yields an optimal selection.

If $p_i' = p_i$, it is straightforward to see that $S^{*\prime} = S^* \cup \{i'\} \setminus \{i\}$ is optimal. Now we consider the case $p_{i'} > p_i$. For any subset $S \subseteq [n]$, we consider the additional variance induced by adding any item $k \notin S$ to the subset S:

$$\begin{aligned}V(S\cup\{k\}) - V(S) &= \mathbb{E}_{Z_{S\cup\{k\}}}\left[\rho\left(\sum_{i\in S\cup\{k\}} Z_i\right) - \rho\left(\sum_{i\in S} Z_i\right)\right] \\ &\overset{\text{(i)}}{=} \mathbb{E}_{Z_S}\left[\mathbb{E}_{Z_k}\,\rho\left(\sum_{i\in S\cup\{k\}} Z_i\right) - \rho\left(\sum_{i\in S} Z_i\right)\right] \\ &\overset{\text{(ii)}}{=} p_k\cdot\underbrace{\mathbb{E}_{Z_S}\left[\rho\left(\sum_{i\in S} Z_i + 1\right) - \rho\left(\sum_{i\in S} Z_i\right)\right]}_{T(S)},\end{aligned} \tag{9}$$

where (i) is true by the assumption that the random variables $\{Z_i\}_{i=1}^n$ are independent, and (ii) is true by taking an expectation over Z_t. Note that the term T defined in (9) is independent from the probability p_k associated with item k. Setting $S = S^* \setminus \{i\}$ and $k \in \{i, i'\}$ in (9), we have

$$V(S^*) - V(S^* \setminus \{i\}) = p_i \cdot T(S^* \setminus \{i\}), \tag{10a}$$
$$V(S^{*\prime}) - V(S^* \setminus \{i\}) = p_{i'} \cdot T(S^* \setminus \{i\}), \tag{10b}$$

Combining (10a) with the assumption that S^* is an optimal selection, we have $T(S^* \setminus \{i\}) \geq 0$. Combining (10) with the assumption that $p_{i'} > p_i$, we have

$$V(S^{*\prime}) \geq V(S^*). \tag{11}$$

Since by assumption S^* is an optimal selection, the equality holds in (11) and $S^{*\prime}$ is also an optimal selection.

If $S^{*\prime}$ is not a prefix, we keep repeating the same modification, until the resulting selection is a prefix. Since $p_{i'} \geq p_i$, we have $i' \leq i$, and hence in each modification, the sum of the indices in the selection decreases, namely $\sum_{k \in S^{*\prime}} k < \sum_{k \in S^*} k$. Hence, the sequence of modifications terminates, yielding an optimal selection that is a prefix.

Step 2: The stopping criterion obtains a best prefix among all prefixes. We now show that the stopping criterion in Algorithm 1 obtains a prefix with the minimum variance among all prefixes. By Step 1, there exists a prefix that is an optimal selection, so this prefix obtained by the stopping criterion is optimal.

We consider the term T in (9) when adding to a selection $S \subseteq [n]$ some new item $k \notin S$. We have

$$\begin{aligned}
T(S \cup \{k\}) &= \mathbb{E}_{Z_S} \mathbb{E}_{Z_k} \left[\rho\left(\sum_{i \in S \cup \{k\}} Z_i + 1 \right) - \rho\left(\sum_{i \in S \cup \{k\}} Z_i \right) \right] \\
&= p_k \mathbb{E}_{Z_S} \left[\rho\left(\sum_{i \in S} Z_i + 2 \right) - \rho\left(\sum_{i \in S} Z_i + 1 \right) \right] + (1 - p_k) \cdot \mathbb{E}_{Z_S} \left[\rho\left(\sum_{i \in S} Z_i + 1 \right) - \rho\left(\sum_{i \in S} Z_i \right) \right] \\
&\overset{\text{(i)}}{\geq} \mathbb{E}_{Z_S} \left[\rho\left(\sum_{i \in S} Z_i + 1 \right) - \rho\left(\sum_{i \in S} Z_i \right) \right] = T(S),
\end{aligned} \tag{12}$$

where step (i) uses the property that $\rho(t+2) - \rho(t+1) \geq \rho(t+1) - \rho(t)$ for any $t \in \mathbb{R}$, due to the convexity of ρ. By the definition of Algorithm 1, it yields a prefix $\{1, 2, \ldots, i^*\}$ such that $T([i]) \leq 0$ for all $i \leq i^*$, and $T([i^* + 1]) > 0$. By (12), it can be verified that $T([i]) > 0$ for all $i > i^*$. Hence, the variance decreases or stays the same for adding each item up to item i^*, and then strictly increases for adding each of item $(i^* + 1)$ to item n. Hence, the prefix $[i^*]$ attains the minimal variance among all prefixes, and hence is an optimal selection. □

B.4 Proof of Proposition 2

Consider any instance (x, p, λ, M) and any constant $c > 0$. It is straightforward to verify that the optimal solution and the solution given by any of the three greedy algorithms is identical for the instance (x, p, λ, M) and the instance $(cx, p, c\lambda, M)$. Hence, it suffices to construct an instance for a fixed value of $\lambda > 0$. We now construct instances for the greedy algorithms separately.

Instance for PGREEDY. Let $M = 1$. We consider an instance consisting of two items:

$$(x_1, p_1) = (0, 1)$$
$$(x_2, p_2) = (1, p),$$

for some $p \in [0, 1)$ whose value is specified later. It is straightforward to derive that PGREEDY selects item 1, attaining an objective of 0. On the other hand, the objective of only picking item 2 is:

$$p - \lambda(1 - p).$$

We take p to be sufficiently large (close to 1) such that $p/(1-p) > \lambda$. Then the objective of only picking item 2 is strictly positive, and hence the objective of the optimal solution is strictly positive.

Instance for XGREEDY and XPGREEDY. Let $M = 1$. We consider an instance consisting of two items:

$$(x_1, p_1) = (1, 1)$$
$$(x_2, p_2) = \left(2 + \epsilon, \frac{1}{2}\right),$$

for some $\epsilon > 0$ whose value is specified later. The objective for the four possible selections is computed as:

$$\begin{aligned} U(\emptyset) &= -\lambda \\ U(\{1\}) &= 1 \\ U(\{2\}) &= 1 + \frac{\epsilon - \lambda}{2} \\ U(\{1,2\}) &= 2 + \frac{\epsilon - \lambda}{2}. \end{aligned}$$

It is straightforward to derive that both XGREEDY and XPGREEDY pick item 2 first followed by item 1, attaining an objective of $2 + \frac{\epsilon - \lambda}{2}$. We set set any value of λ such that $\lambda > 4$, and set $\epsilon = \frac{\lambda}{2} - 2 > 0$. The objective becomes $1 - \frac{\lambda}{4} < 0$. On the other hand, the optimal selection is $S^* = \{1\}$, with a positive objective of 1. □

B.5 Proof of Theorem 1

We first describe a pseudo-polynomial time algorithm in [3] for solving a specific form of rank-1 binary quadratic programming. Then we describe our algorithm, which operates by rounding the parameters and using the pseudo-polynomial time algorithm as a sub-routine.

Pseudo-polynomial time algorithm of [3]. The authors of [3] study unconstrained binary quadratic programming problems of the form

$$\min_{x \in \{0,1\}^n} \langle x, Ax \rangle + \langle b, x \rangle.$$

where $A \in \mathbb{R}^{n \times n}$ is symmetric and $a \in \mathbb{R}^n$. When A has rank one, this can be reformulated as

$$\min_{x \in \{0,1\}^n} f(x) = \min_{x \in \{0,1\}^n} \langle a, x \rangle + \gamma(\beta + \langle u, x \rangle)^2 \tag{13}$$

for some $a, u \in \mathbb{R}^n$ and $\gamma, \beta \in \mathbb{R}$; it is worth noting that there are degrees of freedom in the coefficients in this reformulation.

Their approach depends on the magnitude of the coefficients and scalars of this problem. Then they show the following:

Proposition 3 (Proposition 1 of [3] with $d = 1$)**.** *Consider an instance of* (13) *with* $u \in \mathbb{Z}^n$, $\beta \in \mathbb{Z}$, $a \in \mathbb{Z}^n$, *and* $\gamma \in \mathbb{Q}$. *Let* $K := 2\max(\|u\|_\infty, \|a\|_\infty)$. *Then the minimum objective attained by* (13) *can be computed in* $O(K^4 n^5)$ *time.*

We refer to the algorithm satisfying Proposition 3 as R1UBQPSOLVER, which is described within the proof of Proposition 1 in [3]. R1UBQPSOLVER takes as inputs (a, u, γ, β), and outputs the minimum objective attained by (13). We also note that while R1UBQPSOLVER as stated requires $\gamma \in \mathbb{Q}$, this serves only as a sufficient condition for arguing that arithmetic operations involving γ can be performed efficiently. Since our runtime analysis will be in terms of the number of arithmetic operations performed, we will remain agnostic to the exact representation of the numbers in our problem instance.

Modified R1UBQP Solver. It will be convenient for us to use a slightly more general version of this binary quadratic programming algorithm, which essentially serves as a rescaling of R1UBQPSOLVER. We will call this R1UBQPSOLVER2 (algorithm 3).

Algorithm 3 R1UBQPSOLVER2

Require: $u \in \mathbb{Z}^n$, $\beta \in \mathbb{Z}$, $a \in \mathbb{Q}^n$ such that $a = \frac{a'}{E}$ for some $a' \in \mathbb{Z}^n$ and $E \in \mathbb{N}$, and $\gamma \in \mathbb{R}$
Ensure: $\min_{b \in \{0,1\}^n} \langle a, b \rangle + \gamma(\beta + \langle u, b \rangle)^2$
1: $a' \leftarrow Ea$
2: $\gamma' \leftarrow E\gamma$
3: **return** $\frac{1}{E}$R1UBQPSOLVER(a', γ', β, u)

Proposition 4. *Consider an instance of* (13) *with* $u \in \mathbb{Z}^n$, $\beta \in \mathbb{Z}$, $a \in \mathbb{Q}^n$ *such that* $a = \frac{a'}{E}$ *for some* $a' \in \mathbb{Z}^n$ *and* $E \in \mathbb{N}$, *and* $\gamma \in \mathbb{R}$. *Let* $K' \coloneqq 2\max(\|u\|_\infty, \|a'\|_\infty)$. *Then* R1UBQPSOLVER2 *computes the minimum objective attained by* (13) *in* $O(K'^4 E^4 n^5)$ *time.*

Proof. The correctness of R1UBQPSOLVER2 follows immediately from Proposition 3. For the runtime guarantee, note that the invocation of R1UBQPSOLVER is with $a'_i = Ea_i$, so the guarantee from R1UBQPSOLVER holds with $K \geq EK'$. □

Proposed FPTAS. We now derive the following FPTAS for our problem, which uses R1UBQPSOLVER2 as a subroutine:

Algorithm 4 APPROXL$_2$

Require: Problem instance $\mathcal{I} = (x, p, \lambda, M)$; additive error $\epsilon > 0$
Ensure: $S \subseteq [n]$ for which $U(S) \geq U(S^*) - \epsilon$
1: $D \leftarrow \lceil 2n\lambda(2M + 3(n+1))/\epsilon \rceil$
2: $E \leftarrow \lceil 2n/\epsilon \rceil$
3: $\bar{a} \leftarrow \frac{1}{E}\lfloor E\left(-p \circ x + \lambda \cdot p - \lambda \cdot (p \circ p)\right) \rfloor$
4: $\gamma' \leftarrow \lambda / D^2$
5: $\beta' \leftarrow DM$
6: $u' \leftarrow \lfloor Dp \rfloor$
7: $\overline{OPT} \leftarrow -$R1UBQPSOLVER2$(\bar{a}, \gamma', \beta', u')$
8: $S \leftarrow [n]$
9: **while** $\exists i \in S$ such that $-$R1UBQPSOLVER2$(\bar{a}|_{S\setminus\{i\}}, \gamma'|_{S\setminus\{i\}}, \beta'|_{S\setminus\{i\}}, u'|_{S\setminus\{i\}}) = \overline{OPT}$ **do**
10: $\quad S \leftarrow S \setminus \{i\}$
11: **return** S

Given an instance of our problem, APPROXL$_2$ generates a rounded instance and then runs R1UBQPSOLVER2 on this rounded instance. The objective is guaranteed to be close to optimal, and so APPROXL$_2$ first finds the optimal rounded value, and then identifies a set which attains this rounded value.

We prove that APPROXL$_2$ (Algorithm 4) is a FPTAS for our problem when $\rho = L_2$.

Rewriting the objective in the form of (13)**.** We begin by establishing that if $\rho = L_2$ then $U(S)$ can be written in the form of Eq. (13). Recall from (2) that our objective can be written as

$$U(S) = \sum_{i \in S} p_i x_i - \lambda \cdot \mathbb{E}\left(\sum_{i \in S} Z_i - M\right)^2,$$

and recall from (3) that using the notation $b \in \{0,1\}^n$ where $b_i \coloneqq \mathbb{1}\{i \in S\}$ (and additionally abusing notation to let $U(b) \coloneqq U(S(b))$), we have

$$U(b) = \sum_{i \in [n]} b_i p_i x_i - \lambda \cdot \mathbb{E}\left(\sum_{i \in [n]} b_i Z_i - M\right)^2. \tag{14}$$

For any two vectors $u, v \in \mathbb{R}^n$, let $u \circ v$ denote the entrywise product. Expanding the squared term in (14) yields

$$\begin{aligned}
U(b) &= \sum_{i\in[n]} b_i p_i x_i - \lambda \cdot \left(\mathbb{E}\left(\sum_{i\in[n]} b_i Z_i \right)^2 - 2M \sum_{i\in[n]} b_i p_i + M^2 \right) \\
&\overset{\text{(i)}}{=} (p \circ x)^T b - \lambda \cdot \mathbb{E}\left[\sum_{i\in[n]} b_i Z_i + \sum_{(i,j)\in[n]^2, i\neq j} b_i b_j Z_i Z_j \right] + 2\lambda M \cdot p^T b - \lambda M^2 \\
&= (p \circ x)^T b - \lambda \cdot p^T b - \lambda \cdot \sum_{(i,j)\in[n]^2, i\neq j} b_i b_j p_i p_j + 2\lambda M \cdot p^T b - \lambda M^2 \\
&= (p \circ x)^T b - \lambda \cdot p^T b - \lambda \cdot \left(\left(\sum_{i\in[n]} b_i p_i \right)^2 - \sum_i b_i^2 p_i^2 \right) + 2\lambda M \cdot p^T b - \lambda M^2,
\end{aligned}$$

where step (i) uses the fact that b_i and Z_i are binary, and hence $b_i^2 = b_i$ and $Z_i^2 = Z_i$. Using the fact that b_i is binary again, we have

$$\begin{aligned}
U(b) &= (p \circ x)^T b - \lambda \cdot p^T b - \lambda \cdot (p^T b)^2 + \lambda \cdot (p \circ p)^T b + 2\lambda M \cdot p^T b - \lambda M^2 \\
&= \left(p \circ x - \lambda \cdot p + \lambda \cdot (p \circ p)\right)^T b - \lambda \left(-M + p^T b\right)^2 . \qquad (15)
\end{aligned}$$

Negating (15) yields

$$\min_{S\subseteq[n]} -U(b) = \min_{b\in\{0,1\}^n} \left(-p \circ x + \lambda \cdot p - \lambda \cdot (p \circ p)\right)^T b + \lambda \left(-M + p^T b\right)^2 , \qquad (16)$$

which matches the form of (13) with

$$\begin{aligned}
a &\coloneqq -p \circ x + \lambda \cdot p - \lambda \cdot (p \circ p), \qquad (17) \\
\gamma &\coloneqq \lambda, \\
\beta &\coloneqq -M, \\
u &\coloneqq p.
\end{aligned}$$

Rounding the parameters. We argue that rounding the parameters of an instance does not significantly affect the objective value. Consider rounded probabilities $\bar{p}$, with $|p_i - \bar{p}_i| \leq 1/D$ and rounded a_i with $|a_i - \bar{a}_i| \leq 1/E$, for some integers D and E to be specified later. How much does the value of (13) change for the input (17) if these $u_i = p_i$ are changed to $\bar{u} \coloneqq \bar{p}_i$ and a_i to $\bar{a}_i$, regardless of b? Let $\Psi(b, a, \gamma, \beta, u) = \langle a, b\rangle + \gamma(\beta + \langle u, b\rangle)^2$ for concreteness, with the choice of variables specified in (17). Letting $p_i' \coloneqq p_i - \bar{p}_i$ (for compactness), the difference before and after rounding is bounded by

$$\begin{aligned}
\Delta &= |\Psi(b, a, \gamma, \beta, u) - \Psi(b, \bar{a}, \gamma, \beta, \bar{u})| \\
&\leq \left|(a - \bar{a})^T b\right| + \gamma \left|(p^T b)^2 - (\bar{p}^T b)^2 - 2M(p - \bar{p})^T b\right| \\
&\leq \frac{n}{E} + \gamma \underbrace{\left(\left|(p^T b)^2 - (\bar{p}^T b)^2\right|\right)}_{T} + \gamma \left(2M \frac{n}{D}\right) \qquad (18)
\end{aligned}$$

We bound the term T by

$$\begin{aligned}
T &= \sum_{i \neq j} b_i b_j (p_i p_j - \bar{p}_i \bar{p}_j) + \sum_i b_i^2 (p_i^2 - \bar{p}_i^2) + \\
&\leq \sum_{i \neq j} |p_i p_j - \bar{p}_i \bar{p}_j| + \sum_i |p_i^2 - \bar{p}_i^2| \\
&\leq \sum_{i \neq j} (\bar{p}_i p'_j + \bar{p}_j p'_i + p'_i p'_j) + \sum_i (2\bar{p}_i p'_i + p'^2_i) \\
&\leq \sum_{i \neq j} (p'_j + p'_i + p'_i p'_j) + \sum_i (2p'_i + p'^2_i) \\
&\leq 2n^2 \frac{1}{D} + n^2 \frac{1}{D^2} + n \frac{2}{D} + n \frac{1}{D^2} \\
&= \frac{n}{D} \left(2(n+1) + \frac{n+1}{D} \right)
\end{aligned} \tag{19}$$

Plugging (19) back to (18), we have

$$\Delta \leq \frac{n}{E} + \frac{\gamma n}{D} \left(2(n+1) + \frac{n+1}{D} + 2M \right). \tag{20}$$

Recalling that $\gamma = \lambda$ for our problem. It can be verified by (20) that choosing $D \geq 2n\lambda(2M + 3(n+1))/\epsilon$ and $E \geq 2n/\epsilon$ ensures that $|\Psi(b, a, \gamma, \beta, u) - \Psi(b, a, \gamma, \beta, \bar{u})| \leq \epsilon$, for any arbitrary binary vector b.

We now define rounded versions of the problem parameters, which are rounded to increments of D. For all i, let

$$\begin{aligned}
P_i &\coloneqq \lfloor D p_i \rfloor \\
\bar{u}_i &\coloneqq \frac{P_i}{D} = \frac{1}{D} \lfloor D p_i \rfloor \\
u'_i &\coloneqq P_i = \lfloor D p_i \rfloor,
\end{aligned}$$

and $\beta' \coloneqq -DM$ and $\gamma' \coloneqq \frac{\gamma}{D^2} = \frac{\lambda}{D^2}$. Then letting $S(b)$ be the set indicated by b,

$$\Psi(b, a, \gamma, \beta, u) = \langle a, b \rangle + \gamma(\beta + \langle u, b \rangle)^2 = -U(S(b))$$

by (16), while

$$\begin{aligned}
\Psi(b, \bar{a}, \gamma', \beta', u') &= \langle \bar{a}, b \rangle + \gamma'(\beta' + \langle u', b \rangle)^2 \\
&= \langle \bar{a}, b \rangle + \frac{\gamma}{D^2} (\beta D + \langle u', b \rangle)^2 \\
&= \langle \bar{a}, b \rangle + \gamma \left(\beta + \left\langle \frac{u'}{D}, b \right\rangle \right)^2 \\
&= \Psi(b, \bar{a}, \gamma, \beta, \bar{u}).
\end{aligned} \tag{21}$$

We have just argued that $|U(S(b)) - \Psi(b, \bar{a}, \gamma, \beta, \bar{u})| \leq \epsilon$ when $D \geq 2n\lambda(2M + 3(n+1))/\epsilon$ and $E \geq 2n/\epsilon$. Observe also that the parameters $\bar{a}$, γ', β', u' are such that (21) satisfies the requirements for Proposition 4. Therefore $\overline{OPT} \leftarrow -\text{R1UBQPSOLVER2}(\bar{a}, \gamma', \beta', u')$ is some objective value within $\pm\epsilon$ of our optimal value $U(S^*)$.

Algorithm 4 therefore begins by finding some value $\overline{OPT}$ which is the optimal value of the rounded instance of the problem realized by some $b \in \{0, 1\}^n$ and within ϵ of $-U(S(b)f)$. Since whatever value the b^* corresponding to S^* attains on the rounded instance is within ϵ of $-U(S^*)$, it follows that this b is an additive ϵ-approximation to $U(S^*)$.

The remainder of APPROXL$_2$ is dedicated to reconstructing the set $S(b)$ itself. It does this by iteratively removing candidate components of the solution $i \in [n]$, determining whether or not each is necessesarily part of some such $S(b)$ (subject to the i already discarded).

Runtime. R1UBQPSOLVER2 has runtime $O(K^4E^4n^5)$, and our reduction takes K to be the maximum of D and $\max(\lambda, x_{\max})/E$. In our reduction, $E = O((1+\lambda)n(M+n)/\epsilon)$. Since APPROXL$_2$ makes one call to R1UBQPSOLVER and all other steps are negligible, it therefore runs in time $O(\frac{n^9(M+n)^4(1+\lambda)^4}{\epsilon^4})$.

In the case that $x_i \geq 0$ for all $i \in n$, we assume that $M \leq n$, since for $M \geq n$ it is optimal to take $S = [n]$; in this case the runtime guarantee is therefore $O(\frac{n^{13}(1+\lambda)^4}{\epsilon^4})$. □

B.6 Proof of Theorem 2

Starting from (16), we follow the reduction from SUBSETSUM outlined in Section 2 of [3]. Given an instance of SUBSETSUM, we construct an instance of our problem with the L_2 objective. We construct the p_i from the SUBSETSUM instance weights such that $M = 1$.[2]

An instance of SUBSETSUM is given by a set of natural numbers $(t_1, \ldots, t_n)$ and a target sum T. We assume that $T > 0$, since it is trivial to decide instances where $T = 0$. Again let b be the binary indicator for the set $S(b)$. Then $\sum_{i \in S(b)} t_i = T$ if and only if $(\sum_i b_i t_i - T) = 0$. We assume without loss of generality that $t_i \leq T$, so let us rescale the instance of and take $p_i \coloneqq t_i/T$. Our target is now $M \coloneqq 1$ and these $p_i \in [0, 1]$ are valid probabilities.

We now choose the remaining problem parameters, such that the linear term becomes zero, and the quadratic term becomes the SUBSETSUM problem. Since $a = -p \circ x + \lambda \cdot p - \lambda \cdot (p \circ p)$, we ensure that $\widetilde{c} = 0$ by choosing $x_i \coloneqq \lambda(1 - p_i)$ for all $i \in [n]$. Note that $x_i \geq 0$. The regularizer $\lambda > 0$ can be freely chosen. Then

$$\begin{aligned} U(S_b) &= -\Psi(b, a, \gamma, \beta, u) \\ &= -a^T b - \gamma(\beta + u^T b)^2 \\ &= -\lambda \left(-1 + \sum_i b_i \frac{t_i}{T} \right)^2 \\ &= -\frac{\lambda}{T} \left(-T + \sum_{i \in S(b)} t_i \right)^2 . \end{aligned}$$

Clearly $U(S) \geq 0$ if and only if $\sum_{i \in S} t_i = T$, and so any algorithm for finding the optimal subset S^* for our problem can be used to solve SUBSETSUM, completing the reduction. □

B.7 Proof of Theorem 3

We prove the three parts of the proposition separately. For proving upper bounds on the approximation ratio, we construct "bad" instances. Using the same argument as in the proof of Proposition 2 that the values $\{x_i\}_{i \in [n]}$ may be rescaled according to λ or vise versa, it suffices to construct instances for a fixed value of $\lambda > 0$.

For proving lower bounds for XGREEDY and XPGREEDY (across all instances), it is without loss of generality to assume that all $x_i \geq 0$. Since the loss $\rho = L_1^+$ is monotonic, adding an item always increases the penalty term, and thus adding an item with negative value always decreases the utility. Moreover, all items with negative values form a suffix in the order used by XGREEDY and XPGREEDY, so there are no more items with positive values once the greedy algorithms reach the first negative item. Therefore, XGREEDY and XPGREEDY never choose solutions containing any item with negative value, and hence such items can be ignored for the purposes of these proofs.

B.7.1 Proof of Theorem 3(a)

Let $M = 1$. Consider an instance consisting of two items:

$$\begin{aligned} (x_1, p_1) &= (0, 1) \\ (x_2, p_2) &= \left(1, \frac{1}{2}\right) . \end{aligned}$$

[2] One can instead reduce from EQUIPARTITION by following this construction but ensuring that $M = 2$.

Then PGREEDY selects item 1, attaining an objective of 0. On the other hand, selecting item 2 attains a strictly positive objective of 0.5.

B.7.2 Proof of Theorem 3(b)

We separately prove the upper and lower bounds for XPGREEDY.

Upper bound for XPGREEDY. First, suppose that $p_{\min}$ is such that $1/p_{\min} \in \{2, 3, 4, \ldots\}$. We assume this without loss of generality in order to show that XPGREEDY is $\Omega(p_{\min})$ for any $p_{\min} \in (0, 1]$. This is because for any $p_{\min} \in (1/2, 1]$ the lower bound obtains a constant factor and so is $\Theta(p_{\min})$. And for any choice $p_{\min} \in (0, 1/2)$, consider the instance outlined below with $\frac{1}{\lceil 1/p_{\min} \rceil}$ as the minimum probability, together with an additional item for which $(p_i, x_i) = (p_{\min}, 0)$. Then XPGREEDY never chooses this last item, and the instance below demonstrates an upper bound of $O(\frac{1}{\lceil 1/p_{\min} \rceil}) = O(p_{\min})$.

Our instance is as follows. Let $M = 1$ and $\lambda = \frac{1+2c}{p_{\min}}$, and consider an unlimited number of items from two types:

$$(x_1, p_1) = \left(1 + \frac{c}{2}, 1\right)$$
$$(x_2, p_2) = \left(\frac{1}{p_{\min}}, p_{\min}\right).$$

Note we also assume without loss of generality that $c \leq 1/2$, since if $x_i \leq (1-c) \cdot \lambda$ is true for some $c > 1/2$ then it is true for $c \leq 1/2$ also. Then it can be verified that $x_i \leq (1-c) \cdot \lambda$ for this instance. Specifically, we have

$$(1-c) \cdot \lambda = \frac{1 + c - 2c^2}{p_{\min}} \geq 1 + c - 2c^2 \geq 1 + \frac{c}{2} = x_1$$

and

$$(1-c) \cdot \lambda = \frac{1 + c - 2c^2}{p_{\min}} \geq \frac{1}{p_{\min}} = x_2,$$

for any $c \in (0, 1/2]$.

XPGREEDY adds items one-by-one from the first type. The objective after adding one item is $(1+c/2)$. If a second item is added, the marginal change in the objective is $1 + \frac{c}{2} - \lambda = 1 + \frac{c}{2} - \frac{1+2c}{p_{\min}} < 0$. Hence, XGREEDY selects exactly one item from the first type, attaining an objective of $(1 + c/2)$.

Now consider a selection S consisting of t items of the second type. We now show that for some choice of t, the objective attained by S is $U(S) = \Omega(\frac{1}{p_{\min}})$. We have

$$\begin{aligned} U(S) &= R(S) - \lambda \cdot V(S) \\ &= t - \lambda \cdot \mathbb{E}\Big(|S_Z| - 1\Big)_+ \\ &= t - \lambda\Big(\mathbb{E}|S_Z| - 1 + \mathbb{P}(|S_Z| = 0)\Big) \\ &= t - \lambda\left(t \cdot p_{\min} - 1 + (1 - p_{\min})^t\right). \end{aligned}$$

Choosing $t = \frac{1}{p_{\min}}$, we have

$$\begin{aligned} U(S) &= \frac{1}{p_{\min}} - \lambda\left(\frac{1}{p_{\min}} \cdot p_{\min} - 1\right) - \lambda\left(1 - p_{\min}\right)^{\frac{1}{p_{\min}}} \\ &= \frac{1}{p_{\min}} - \lambda\left(1 - p_{\min}\right)^{\frac{1}{p_{\min}}} \\ &\geq \frac{1}{p_{\min}} - \frac{\lambda}{e}. \end{aligned}$$

Substituting in λ, this becomes

$$\begin{aligned} U(S) &\geq \frac{1}{p_{\min}} - \frac{1}{e} \cdot \frac{1+2c}{p_{\min}} \\ &= \frac{1}{p_{\min}} \cdot \frac{e-1-2c}{e}, \end{aligned}$$

which is $\Omega(1/p_{\min})$ since $c \leq 1/2$.

Therefore we have an instance for which $U(S_{\text{XP}}) \leq c' \cdot p_{\min} U(S)$ for some constant c', establishing an upper bound of $O(p_{\min})$ on the approximation ratio.

Lower bound for XPGREEDY. First, if the total number of items is at most M, then it can be verified that selecting all items is optimal.

We denote by S_M be the M items with highest expected values $x_i p_i$, and denote by S_{XP} the solution that XPGREEDY finds. Moreover, we have $S_M \subseteq S_{\text{XP}}$, because all values are assumed nonnegative, and the penalty term for the L_1^+ loss is zero when adding the first M items. By definition, XPGREEDY only improves the objective in each step. Hence, we have

$$U(S_M) \leq U(S_{\text{XP}}). \tag{22}$$

We now provide a lower bound for the selection S_M. Applying the Mean Bound (Lemma 2) with $\epsilon = \min(c, 1/2)$, we have either

$$|S^*| \leq \frac{c'}{p_{\min}} \tag{23a}$$

where c' is a constant, or

$$\mu^* \leq \frac{101}{100} M. \tag{23b}$$

If (23a) holds, we have $|S^*| \leq \frac{cM}{p_{\min}}$ because $M \geq 1$. If (23b) holds, we have $p_{\min} \cdot |S^*| \leq \mu^* \leq \frac{101}{100} M$, and hence $|S^*| \leq \frac{101}{100} \cdot \frac{M}{p_{\min}}$. Combining the two cases, we have

$$|S^*| \lesssim \frac{M}{p_{\min}}. \tag{24}$$

Next, we consider the expected reward $R(S_M)$ for the selection S_M. We note that $|S^*| \geq |S_M| = M$ by the optimality of S^*. This is because $x_i \geq 0$, so adding any item to a selection containing less than M items only increases the objective. Recall that the selection S_M consists of the M items with the maximum expected reward $p_i x_i$. Hence, the mean expected reward $p_i x_i$ for the set S_M (over all items in this set) is greater than or equal to the mean expected reward for the set S^*. Namely,

$$\begin{aligned} \frac{1}{M} R(S_M) &= \frac{1}{|S_M|} \sum_{i \in S_M} p_i x_i \\ &\geq \frac{1}{|S^*|} \sum_{i \in S^*} p_i x_i = \frac{1}{|S^*|} R(S^*). \end{aligned}$$

Hence, we have

$$\begin{aligned} U(S_M) = R(S_M) &\geq \frac{M}{|S^*|} R(S^*) \\ &\overset{\text{(i)}}{\gtrsim} p_{\min} \cdot R(S^*) \geq p_{\min} \cdot U(S^*), \end{aligned} \tag{25}$$

where step (i) is true by plugging in (24). Combining (25) with (22), we have

$$U(S_{\text{XP}}) \geq U(S_{\text{XP}}) \gtrsim p_{\min} \cdot U(S^*),$$

completing the proof of the lower bound $\Omega(p_{\min})$ of the approximation ratio for XPGREEDY. □

B.7.3 Proof of Theorem 3(c)

We separately prove the upper and lower bounds for XGREEDY.

Upper bound for XGREEDY. Let $M = 1$ and $\lambda = \frac{2}{p}$. Consider an instance consisting of an unlimited number of items from two types:

$$(x_1, p_1) = (1, p_{\min})$$
$$(x_2, p_2) = (1 - \epsilon, 1),$$

where $\epsilon \in (0, 1)$ is a constant, and $p_{\min} \in (0, 1)$. We again suppose without loss of generality that $c \leq 1/2$, and it can be verified that $x_i \leq (1 - c) \cdot \lambda$ for both types of items.

XGREEDY adds items one-by-one from the first type. The objective after adding one item is $p_{\min}$. If a second item is added, the marginal change in the objective is $p_{\min} - \lambda p_{\min}^2 = -p_{\min} < 0$. Hence, XGREEDY selects exactly one item from the first type, attaining an objective of $p_{\min}$.

On the other hand, choosing a single item from the second type attains an objective value of $(1 - \epsilon)$. Therefore XGREEDY has a worst-case approximation ratio of at most $\frac{p_{\min}}{1-\epsilon}$, namely $O(p_{\min})$.

Lower bound for XGREEDY. We modify the construction of S_M in the proof of part (b) to be the set of M items with the highest values x_i, Then we apply similar arguments as in part (b), and outline the steps as follows.

Denote by S_M the set of M items with the highest values x_i, and denote by S_{X} the solution that XGREEDY finds. Then again we have $S_M \subseteq S_{\text{X}}$ and hence

$$U(S_M) \leq U(S_{\text{X}}). \tag{26}$$

We now provide a lower bound for the selection S_M. Using the same argument as in part (b), we have (cf. (24)):

$$|S^*| \lesssim \frac{M}{p_{\min}}. \tag{27}$$

We note that $|S^*| \geq |S_M| = M$ by the optimality of S^*. Since the selection S_M consists of the M items with the maximum values x_i, the mean value for the set S_M is greater than or equal to the mean value for the set S^*. Namely,

$$\frac{1}{M} \sum_{i \in S_M} x_i \geq \frac{1}{|S^*|} \sum_{i \in |S^*|} x_i.$$

Next note that for any $i \in S_M$, $\frac{1}{p_{\min}}(p_i x_i)$ is larger than $p_j x_j$ for any $j \in S^* \setminus S_M$, since for such i and j we have $\frac{1}{p_{\min}}(p_i x_i) \geq x_i \geq p_j x_j$. Therefore,

$$\begin{aligned} U(S_M) = R(S_M) = \sum_{i \in S_M} p_i x_i &\geq p_{\min} \sum_{i \in S_M} x_i \\ &\geq p_{\min} \cdot \frac{M}{|S^*|} \sum_{i \in S^*} x_i \\ &\geq p_{\min} \cdot \frac{M}{|S^*|} \sum_{i \in S^*} p_i x_i \\ &\overset{\text{(i)}}{\gtrsim} p_{\min}^2 \cdot R(S^*) \geq p_{\min}^2 \cdot U(S^*), \end{aligned} \tag{28}$$

where step (i) is true by plugging in (27). Combining (26) with (28) completes the proof of the lower bound $\Omega(p_{\min}^2)$ of the approximation ratio for XGREEDY. □

B.8 Proof of Theorem 4

Notation. We begin with some notation that is used in the proofs in this section. Given any instance $\{x_i, p_i\}_{i \in [n]}$, we construct a *rounded* instance $\{y_i, q_i\}_{i \in [n]}$ as follows. First we round *up* the

probabilities p_i to $q_i := 2^{\lceil \log_2 p_i \rceil}$, that is, the smallest power of two that is greater than or equal to p_i. Then we construct new values y_i such that the expected value of each item is preserved. Formally,

$$q_i \coloneqq \min\left\{\frac{1}{2^i}, i \in \mathbb{N} : \frac{1}{2^i} \geq p_i\right\},$$
$$y_i \coloneqq \frac{p_i}{q_i} x_i.$$

We slightly abuse the notation, and for any selection $S = \{x_i, p_i\}_{i\in[n]}$, we denote by $S' := \{y_i, q_i\}_{i\in[n]}$ the corresponding set with rounded probabilities and values. The parameters M and λ for this rounded instance remain unchanged. Note that by construction, we have

$$R(S) = R(S') \tag{29a}$$
$$V(S) \leq V(S') \tag{29b}$$
$$U(S) \geq U(S'), \tag{29c}$$

Eq. (29a) holds by the definition of the rounded set S'; Eq. (29b) in fact holds for all nondecreasing loss function ρ, because $\sum_{i\in S'} Z_i$ stochastically dominates $\sum_{i\in S} Z_i$; Eq. (29c) follows by combining (29a) and (29b).

Finally, recall the observation from Section 3.2.2 that we assume without loss of generality that $x_i > 0$ for all $i \in [S]$, since the marginal contribution of any i for which $x_i \leq 0$ to any $U(S)$ is nonpositive.

Overview of Algorithm 2. We begin by reiterating the overview of Algorithm 2 presented in Section 3.2.2. At a high level, this algorithm proceeds first by dividing the items into three groups according to their values x_i.

$$N_{\mathrm{L}} \coloneqq \{i \in [n] : x_i \leq (1-\epsilon)\lambda\}$$
$$N_{\mathrm{M}} \coloneqq \{i \in [n] : (1-\epsilon)\lambda < x_i < \lambda\}$$
$$N_{\mathrm{H}} \coloneqq \{i \in [n] : x_i \geq \lambda\}.$$

Since U is submodular (see Lemma 1), the optimal solution within at least one of these groups is constant-competitive with $U(S^*)$. We consider each group separately, and obtain a constant-factor approximation for each group. We now provide an overview of the three cases. In particular, the small items in N_{L} are handled by Algorithm 5, and the medium items in set N_{M} are handled by Algorithm 6.

Algorithm 5 LowValueL$_1$ (with universal constant c)

Require: Problem instance $\mathcal{I} = (x, p, \lambda, M)$, with $x_i \leq (1 - \frac{p_{\min}}{4})\lambda$
Ensure: $S \subseteq [n]$ for which $U(S)/U(S^*) \gtrsim 1$

$\tau \leftarrow \frac{c}{p_{\min}^2} \max\left\{1, \log\left(\frac{1}{p_{\min}}\right), \log\left(\frac{\lambda}{x_{\max}}\right)\right\}$
$\mathcal{L} \leftarrow \{S \subseteq [n] : |S| \leq \tau\}$ // Brute-force small instances
for $S \in \mathcal{L}$ **do**
  Calculate $U(S)$
$S_L \leftarrow \arg\max_{S\in\mathcal{L}} U(S)$
Let q be the rounded p and $Q \leftarrow \{q_i\}$ the distinct rounded probabilities; let t_r be the multiplicity of each rounded probability r in the vector q. // Round large instances
$\mathcal{H} \leftarrow \emptyset$
for $s \in \prod_{r\in Q}\{0, 1, \ldots, t_r\}$ **do**
  Construct S from the s_r many $i \in [n]$ of highest x_i for which $q_i = r$, for each $r \in Q$
  Calculate $U(S)$, using unrounded probabilities and values
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{S\}$
$S_H \leftarrow \arg\max_{S\in\mathcal{H}} U(S)$
return $S \in \{S_L, S_H\}$ maximizing $U(S)$

Algorithm 6 MEDIUMVALUEL$_1^+$

Require: Problem instance $\mathcal{I} = (x, p, \lambda, M)$, with $(1 - \frac{p_{\min}}{4}) \cdot \lambda \leq x_i \leq \lambda$
Ensure: $S \subseteq [n]$ for which $U(S)/U(S^*) = \Omega(1)$
1: **if** $n \leq \frac{36}{p_{\min}^2}$ **then**
2: | **return** $\arg\max_{S \subseteq [n]} U(S)$
3: **else if** $\mu_{[n]} \geq M$ **then**
4: | Choose any $S \subseteq [n]$ such that $M \leq \mu_S < M + 1$
5: **else**
6: | Choose any $S \subseteq [n]$ such that $\frac{\mu_{[n]}}{3} \leq \mu_S \leq \frac{\mu_{[n]}}{2}$
7: **return** S

- **Low-value items $N_{\mathbf{L}}$ (Algorithm 5):** LOWVALUEL$_1^+$ presented in Algorithm 5 handles the case where items have low values. It consists of two parts: a search over small candidate solutions and a search over rounded candidate solutions. In the first part, we brute-force all small solutions whose size are at most τ (Line 3). This brute-force search succeeds if the optimal selection is small.

  The second part is the technical crux of proving the constant-factor approximation of LOWVALUEL$_1^+$. In the second part, we compute rounded probabilities and values (q_i, y_i) for each item. This rounding procedure reduces the number of candidate solutions dramatically. We then brute-force over all rounded solutions (Line 8), select the rounded solution that maximizes the objective value, and prove that this solution is comparable to the (unrounded) optimal solution. Since the first part succeeds the case where the optimal solution is small, we may assume in this second part that the optimal solution is sufficiently large; this allows us to prove that our selection is robust to rounding.

  As an aside, we take the rounding to be to powers of two, but our analysis generalizes to rounding to powers of $(1 + c)$ for any constant $c > 0$, and this parameter may be tuned in order to trade off between runtime and performance in practice.

- **Medium-value items $N_{\mathbf{M}}$ (Algorithm 6):** MEDIUMVALUEL$_1^+$ presented in Algorithm 6 handles items with values close to λ. If the number of items is small, it brute-forces over all possible solutions (Line 2). If the number of items is large, the algorithm chooses any subset such that the expected number of accepted items is around M (Line 4). If no such subset exists, then the expected number of accepted items when choosing all items must be less than M. In this case, then we choose a subset with approximately half the expected realizations compared to that of all items (Line 6). We choose a proportion less than one in order to ensure that the penalty incurred is not too large relative to the reward. This subset (Line 6) along with the subset defined in (Line 4) and always exists, formalized in the proof of Lemma 4 in Appendix B.8.2.

- **High-value items $N_{\mathbf{H}}$:** for the group of items with values above λ, it is easy to see that choosing the entire group is optimal.

We now prove the approximation ratio and runtime for ONESIDEDL$_1^+$.

Proof of Theorem 4. To begin, we split the items in the optimal set S^*, according to their values:

$$S_L^* \coloneqq S^* \cap N_L,$$
$$S_M^* \coloneqq S^* \cap N_M,$$
$$S_H^* \coloneqq S^* \cap N_H.$$

By the submodularity of $U(S)$ in Lemma 1, we have $U(S_L^*) + U(S_M^*) + U(S_H^*) \geq U(S^*)$. In particular, this implies that $\max\{U(S_L^*), U(S_M^*), U(S_{\mathrm{H}}^*)\} \geq \frac{1}{3} U(S^*)$. In order to provide a constant-factor approximation to $U(S^*)$, it therefore suffices to identify sets which provide constant-factor approximations to $U(S_L^*)$, $U(S_M^*)$, and $U(S_H^*)$, and return the set with the highest objective value among them. We choose $\epsilon \coloneqq p_{\min}/4$ to determine the boundary between N_L and N_M, and address each group separately. In each case we seek to find a subset which competes with the optimal subset of N_L (say), which in turn is an approximation to $U(S_L^*)$.

Low-value items $N_{\mathbf{L}}$. The following lemma provides the approximation guarantee of LOWVALUEL_1^+.

Lemma 3 (Small x_i)**.** *Suppose that $x_i \le (1 - \frac{p_{min}}{4}) \cdot \lambda$ for all $i \in [n]$. Then* LOWVALUEL_1^+ *(Algorithm 5) is a constant-factor approximation to $U(S^*)$ which runs in time $n^{\frac{c}{p_{min}^2}\max\left\{1,\log\left(\frac{1}{p_{min}}\right),\log\left(\frac{\lambda}{x_{max}}\right)\right\}}$, where c is a universal constant.*

The proof of this lemma is provided in Appendix B.8.1, and is arguably the heart of the analysis of ONESIDEDL_1^+. By applying this lemma to $[n] = N_L$ we obtain S_L with $U(S_L)$ within a constant factor to the optimal objective among selections within N_{L}, and hence a constant factor to $U(S_L^*)$.

Medium-value items $N_{\mathbf{M}}$. The following lemma provides the approximation ratio guarantee of MEDIUMVALUEL_1^+.

Lemma 4 (Medium x_i)**.** *If $\lambda\left(1 - \frac{p_{min}}{4}\right) \le x_i \le \lambda$ for all $i \in [n]$ then* MEDIUMVALUEL_1^+ *(Algorithm 6) finds some $S \subseteq [n]$ which is a constant-factor approximation to $U(S^*)$ and runs in time $n^{O(1/p_{min}^2)}$.*

The proof of this lemma is provided in Appendix B.8.2. By applying this lemma to $[n] = N_M$ we obtain S_M with $U(S_M)$ within a constant factor to the optimal objective among all selections within N_{M}, and hence a constant factor to $U(S_M^*)$.

High-value items $N_{\mathbf{H}}$. This case is simple: we select all items by taking $S_H = N_H$. It can be verified that adding every item strictly increases the objective, and hence N_{H} attains the optimal objective among all selections within N_{H}. By the optimality of S_H, we have $U(S_H) = U(S_H^*)$.

Putting the three cases together, we have S_L, S_M, and S_H, and by the argument provided above at least one of these is a constant-factor approximation to $U(S^*)$. Therefore choosing the one with highest objective value gives a constant-factor approximation.

Runtime. The algorithms for the cases above operate by identifying a collection of sets to test the objective value of, and then evaluating the objective. Fortunately this can be done efficiently.

Lemma 5 (Efficient Objective Evaluation)**.** *Suppose that $\rho(a, M)$ is known for all $a \in \{0, 1, \ldots, n\}$. For any set of items $\{x_i, p_i\}_{i\in[n]}$, the objective $U([n])$ can be computed in $O(n^2)$ arithmetic operations.*

This is proved in Appendix B.8.3. Applying this lemma to any candidate subset S shows that the objective with respect $S \subseteq [n]$ can be computed in $O(|S|^2)$ arithmetic operations.

By Lemma 3, the runtime of LOWVALUEL_1^+ is $n^{\frac{c}{p_{\min}^2}\max\left\{1,\log\left(\frac{1}{p_{\min}}\right),\log\left(\frac{\lambda}{x_{\max}}\right)\right\}}$. By Lemma 4 the runtime of MEDIUMVALUEL_1^+ is $n^{O(1/p_{\min}^2)}$, which is less than that of LOWVALUEL_1^+. Finally, the high-value items case entails evaluating the objective in of a single set; by Lemma 5 this can be done in $O(n^2)$.

The cost of combining these cases is polynomial in n, and so the brute force stage of LOWVALUEL_1^+ dictates the runtime of ONESIDEDL_1^+, giving the claimed runtime of $n^{c\frac{1}{p_{\min}^2}\max\left\{1,\log\left(\frac{1}{p_{\min}}\right),\log\left(\frac{\lambda}{x_{\max}}\right)\right\}}$ for some universal constant $c > 0$. □

We now turn to the statements and proofs of the supporting lemmas.

The following lemma says that the solution can be downsampled so that its $\sum_i p_i$ is at most a constant factor from the original, while $\sum_i p_i x_i$ is at least a constant factor from the original. Informally, we simply select the appropriate number of items with the highest x_i.

Lemma 6 (Downsampling Lemma)**.** *Consider an instance $\{p_i, x_i\}_{i\in[n]}$ with $x_i \ge 0$ for all $i \in [n]$. Then for any $S \subseteq [n]$ and any $\beta \in [0, 1]$, there exists some $T \subseteq S$ that satisfies*

$$\mu_T \le \beta \cdot \mu_S \tag{30a}$$

and

$$R(T) \geq \beta \left(1 - \frac{1}{\beta \cdot p_{min} \cdot |S|}\right) \cdot R(S). \tag{30b}$$

The proof of this lemma is provided in Appendix B.8.4. If we were allowed to choose items to be in T fractionally, then condition (30b) would more closely mimic condition (30a) and the proof of this lemma would be even more straightforward; as it is, condition (30b) must be slightly weaker since we must sometimes leave the last item out of T in order to satisfy condition (30a).

This lemma supports the efficient search over rounded solutions which is conducted in LOWVALUEL$_1^+$. Informally, it does this by proving that if some starting set satisfies certain properties, then either there is a small subset with good objective value, or the search over rounded solutions will identify a subset with good objective value.

Lemma 7 (Rounding Lemma)**.** *Consider the one-sided $\rho = L_1^+$ loss. Consider any selection $S \subseteq [n]$ that simultaneously satisfies*

$$U(S) \geq 0 \tag{31a}$$

$$\mu_S \leq \frac{3}{2}M \tag{31b}$$

$$\lambda V(S) \leq \frac{1}{15}R(S). \tag{31c}$$

Then there exists some subset $T \subseteq S$ that satisfies either

$$U(T) \geq \frac{1}{3}U(S) \qquad \textit{and} \qquad |T| \leq \frac{24}{p_{min}}, \tag{32a}$$

or

$$U(T) \geq U(T') \geq \frac{1}{24}U(S), \tag{32b}$$

where T' denotes the rounded instance of the set T.

The proof of this lemma is provided in Appendix B.8.5.

This next lemma bounds the penalty of a subset with expected realized size smaller than M. It uses the independence of the events Z_i to apply tail bounds to the probability that the realized size of the subset exceeds M. When applied to a downsampled subset derived from Lemma 6, it will show that the penalty decreases exponentially while the reward decreases only linearly, yielding a subset which is within a small factor of the starting set's objective but is much less balanced.

Lemma 8 (Downsampling Penalty Bound)**.** *Consider the one-sided $\rho = L_1^+$ loss. Consider any selection $S \subseteq [n]$ such that $\mu_S \leq M$. Then for all $k \in \mathbb{N}_+$, the penalty term is bounded as*

$$V(S) \leq \lambda \cdot k \cdot e^{\frac{-2(M-\mu_S)^2}{|S|}} \cdot \left(1 - e^{\frac{-4(M-\mu_S)k}{|S|}}\right)^{-2}. \tag{33}$$

The proof of this lemma is provided in Appendix B.8.6. Informally, under this loss the penalty increases linearly in the extent to which the realized size of S exceeds M, while the probability that such a violation occurs decreases exponentially. The parameter k is the size of the buckets for which the analysis of these competing influences is performed.

B.8.1 Proof of Lemma 3

Recall that we define $\tau := \frac{c}{p_{\min}^2} \max\left\{1, \log\left(\frac{1}{p_{\min}}\right), \log\left(\frac{\lambda}{x_{\max}}\right)\right\}$ in Line 2 of Algorithm 5. Let c_0 be the universal constant identified in Lemma 2. With $p_{\min} \leq 1$, it is straightforward to verify that there exists a universal constant c, such that τ is bounded by

$$\tau > \frac{c_0}{p_{\min}} \log\left(\frac{4}{p_{\min}}\right), \tag{34a}$$

$$\tau \geq \frac{24}{p_{\min}} \tag{34b}$$

$$\tau \geq \frac{9}{2}\left(\frac{1}{p_{\min}^2}\left(7 + \log\left(\frac{1}{p_{\min}}\right) + 3\log\left(\frac{\lambda}{x_{\max}}\right)\right)\right) \tag{34c}$$

We use these bounds in the remaining proof.

In Line 2-5, we evaluate the objective for each selection S with $|S| \leq \tau$ by brute-force. Hence, if $|S^*| \leq \tau$, then the optimal selection is correctly identified. It remains to consider the case when $|S^*| > \tau$.

When $|S^*| > \tau$, we apply the Mean Bound (Lemma 2) with $\epsilon = \frac{p_{\min}}{4}$. We have either

$$|S^*| \leq \frac{c_0}{p_{\min}} \log\left(\frac{4}{p_{\min}}\right) \tag{35a}$$

or

$$\mu^* \leq \frac{101}{100} M. \tag{35b}$$

The bound (34a) on τ contradicts (35a). Hence we have that (35b) holds, namely $\mu^* \leq \frac{101}{100} M$ from.

We call a set S "balanced" if $\lambda V(S) > \frac{1}{15} R(S)$, that is, the penalty term is a nontrivial portion of the reward term. Otherwise, we call the set "unbalanced". We consider the following two cases separately depending on whether the set S^* is balanced or not.

Case 1: $|S^*| > \tau$ and $\lambda V(S^*) \leq \frac{1}{15} R(S^*)$.

Note that the optimal selection always has a nonnegative objective for the L_1^+ loss. That is, $U(S^*) \geq 0$. Hence, conditions (31) are satisfied. Applying the Rounding Lemma (Lemma 7), there exists some $T \subseteq S^*$ such that

$$U(T) \geq \frac{1}{3} U(S^*) \qquad \text{and} \qquad |T| \leq \frac{24}{p_{\min}}, \tag{36a}$$

or

$$U(T) \overset{\text{(i)}}{\geq} U(T') \geq \frac{1}{24} U(S^*), \tag{36b}$$

In this first case (36a), by the bound (34b) on τ, we have

$$\tau \geq \frac{24}{p_{\min}} \geq |T|.$$

Hence, the selection T is included in the brute-force search. We obtain a constant-factor approximation to $U(S^*)$ in the brute-force search over small solutions (Line 5 of Algorithm 5).

In the second case (36b), if $|T| \leq \tau$, then again the selection T is included in the brute-force search in Line 5 of Algorithm 5, and the brute-force identifies a solution which is at least as good and hence a constant-factor approximation. If $|T| > \tau$, then Line 6 to Line 12 search over all possible rounded solutions, including T' which is a constant-factor approximation due to (36b). Hence, it identifies a solution which is at least as good and hence a constant-factor approximation. identifies some $\widehat{T}$ for which $U(\widehat{T}) \geq U(\widehat{T}') \geq U(T')$, which provides a constant-factor approximation to $U(S^*)$.

Case 2: $|S^*| > \tau$ and $\lambda V(S^*) > \frac{1}{15} R(S^*)$.

As an overview of this case, we appeal to the Downsampling Lemma (Lemma 6) with a small downsampling ratio in order to obtain some $T \subseteq S$, and then argue that $\lambda V(T) \leq \frac{1}{15} R(T)$. Then we obtain a constant-factor approximation to $U(T)$ by solving the rounded problem in Case 1.

Downsampling to an unbalanced set. Starting with the optimal selection S^*, we apply the Downsampling Lemma (Lemma 6) construction some $T \subseteq S^*$ such that this T is unbalanced but still yields a large objective. Specifically, applying Lemma 6 with $\beta = \frac{1}{2}$, there exists some $T \subseteq S^*$ that satisfies $\mu_T \leq \frac{\mu^*}{2}$ and

$$R(T) \geq \left(\frac{1}{2} - \frac{1}{|S^*| \cdot p_{\min}}\right) R(S^*). \tag{37}$$

We assume that the selection T is sufficiently unbalanced as:

$$V(T) \leq \frac{R(T)}{15}. \tag{38}$$

We now identify a constant-factor approximation by similar arguments as in Case 1. Specifically, under the assumption (38), the objective of T satisfies

$$U(T) \overset{\text{(i)}}{\geq} \frac{14}{15} R(T) \overset{\text{(ii)}}{\geq} \frac{14}{15} \cdot \left(\frac{1}{2} - \frac{1}{|S^*| \cdot p_{\min}} \right) R(S^*) \geq \frac{14}{15} \cdot \left(\frac{1}{2} - \frac{1}{|S^*| \cdot p_{\min}} \right) U(S^*), \tag{39}$$

where step (i) is due to the assumption (38), and step (ii) is due to (37). By the assumption $|S^*| \geq \tau$ and the bound (34b) on τ, we have

$$|S^*| \geq \tau \geq \frac{24}{p_{\min}}. \tag{40}$$

Applying (40) to inequality (39), the selection T is a constant-factor approximation to S^* with $U(T) \geq \frac{1}{4} U(S^*)$. Since T is sufficiently unbalanced by assumption (38), using the same arguments as in Case 1 to the set T identifies a selection that is a constant-factor approximation to T, and therefore to $U(S^*)$. It now remains to prove (38).

Proving (38)**.** Recall from (35b) that $\mu^* \leq \frac{101}{100} M$. Hence, we have $\mu_T \leq \frac{\mu^*}{2} \leq \frac{101}{200} M < M$. Then we provide an upper bound on the variance term $V(T)$ by applying Lemma 8 to the set T. Applying Lemma 8 with $k = 1$, we have

$$V(T) \leq \lambda \cdot \underbrace{\text{Exp}\left(\frac{-2(M - \mu_T)^2}{|T|} \right)}_{T_1} \cdot \underbrace{\left(1 - e^{\frac{-4(M-\mu_T)}{|T|}} \right)^{-2}}_{T_2}. \tag{41}$$

We bound the two terms in (41) separately.

Recall from (35b) that $\mu^* \leq \frac{101}{100} M$. We then have

$$M - \mu_T \geq \left(\frac{100}{101} - \frac{1}{2} \right) \mu^* > \frac{1}{3} \mu^*. \tag{42}$$

Using (42), we bound the term T_1 as

$$T_1 = \text{Exp}\left(\frac{-2(M - \mu_T)^2}{|T|} \right) \leq \text{Exp}\left(-\frac{2(\mu^*)^2}{9|T|} \right) \overset{\text{(i)}}{\leq} \text{Exp}\left(-\frac{2}{9} p_{\min}^2 \cdot |S^*| \right) \overset{\text{(ii)}}{\leq} \frac{1}{720} \frac{p_{\min}^3 x_{\max}}{\lambda}$$

where step (i) is true by plugging in $\mu^* \geq |S^*| \cdot p_{\min}$, and $|S^*| \geq |T|$ due to $T \subseteq S^*$; step (ii) is true by the fact that $|S^*| \geq \tau$ with (34c). Let i_{max} be the item with the highest value. With $M \geq 1$, the utility for selecting the item with the highest value is $U(\{i_{\max}\}) = R(\{i_{\max}\}) = p_{i_{\max}} x_{\max} \geq p_{\min} x_{\max}$. Hence, the reward of the optimal selection is bounded by $R(S^*) \geq U(S^*) \geq U(\{i_{\max}\}) \geq p_{\min} x_{\max}$. Hence,

$$T_1 \leq \frac{p_{\min}^2}{45} \frac{R(S^*)}{\lambda}, \tag{43}$$

Using again $M - \mu_T \geq \frac{1}{3} \mu^* \geq \frac{1}{3} \cdot |S^*| \cdot p_{\min}$ and $|S^*| \geq |T|$, we bound the term T_2 as

$$T_2 = \left(1 - e^{\frac{-4(M-\mu_T)}{|T|}} \right)^{-2} \leq \left(1 - e^{\frac{-4\mu^*}{9|S^*|}} \right)^{-2} \leq (1 - e^{-\frac{4}{9} p_{\min}})^{-2} \leq \frac{16}{p_{\min}^2}, \tag{44}$$

where step (i) is true because it can be shown by algebra that

$$1 - e^{-\frac{4}{9}p} - \frac{1}{4} p \geq 0 \qquad \text{for every } p \in [0, 1].$$

Plugging term T_1 from (43) and term T_2 from (44) back to (41) yields

$$V(T) \leq \lambda \cdot \frac{1}{45} R(S^*) \overset{\text{(i)}}{\leq} \frac{1}{15\lambda} \left(\frac{1}{2} - \frac{1}{|S^*| \cdot p_{\min}} \right) R(S^*) \overset{\text{(ii)}}{\leq} \frac{R(T)}{15\lambda}, \tag{45}$$

where step (i) is true by $|S^*| \geq \tau \geq \frac{24}{p_{\min}}$ due to (34b), and step (ii) is true due to (37), proving (38).

Runtime. We conclude by analyzing the runtime of LOWVALUEL$_1^+$.

The number of sets S such that $|S| \le \tau$ is bounded by $|\mathcal{L}| = 2^\tau \le n^\tau$. For each $S \in \mathcal{L}$, we compute $U(S)$ in $O(\tau^2)$. By Lemma 5, the objective of each such set may be evaluated in $O(\tau^2)$ operations, and so the runtime of evaluating the objective for all of these small subsets is $n^{O(\tau)}$.

We also compute the objective for the $O(n^{|Q|})$ rounded sets identified in Algorithm 5, where $|Q| \le \left\lceil \log_2(\frac{1}{p_{\min}}) \right\rceil$, which again by Lemma 5 can be done in $O(n^2)$ operations per set. This is therefore $n^{O(\tau)}$ also.

All of the other simple steps of LOWVALUEL$_1^+$ are also polynomial in n or τ. Therefore, its overall runtime is $n^{O(\tau)}$. □

B.8.2 Proof of Lemma 4

First, we observe that when $n \le \frac{36}{p_{\min}^2}$, we find the optimal solution exactly by brute forcing over all possible solutions $S \subseteq [n]$ (Line 2 of Algorithm 6). Hence, in the rest of the proof we assume that $n \ge \frac{36}{p_{\min}^2}$. We discuss the two cases of $\mu_{[n]} \le M$ (Line 6) and $\mu_{[n]} > M$ (Line 4) separately.

We start by establishing a reformulation of the objective under L_1^+ penalty which is convenient when all items have value close to λ, and a pair of upper and lower bounds.

Bounding the objective. Recall that our objective is of the form $U(S) = \mathbb{E}_Z[F_S(Z)]$, where the random vector $Z \in \{0,1\}^n$ is the Bernoulli realization of each item, and $F_S(Z)$ is the realized utility:

$$\begin{aligned} F_S(Z) &:= \sum_{i\in S} Z_i x_i - \lambda \left(\sum_{i\in S} Z_i - M \right)_+ \\ &= \sum_{i\in S} Z_i x_i - \lambda \cdot \max\left(0,\ \sum_{i\in S} Z_i - M\right) \\ &= \min\left(\sum_{i\in S} Z_i x_i,\ \sum_{i\in S} Z_i x_i - \lambda \cdot \left(\sum_{i\in S} Z_i - M \right) \right). \end{aligned} \tag{46}$$

Plugging in the assumption that $(1-\epsilon)\cdot\lambda \le x_i \le \lambda$ to (46), and recalling that $|S_Z| := \sum_{i\in S} Z_i$, we derive the upper bound

$$\begin{aligned} F_S(Z) &\le \min\left\{ \lambda \cdot |S_Z|, \lambda \cdot M + \sum_{i\in S} Z_i(x_i - \lambda) \right\} \\ \frac{F_S(Z)}{\lambda} &\le \min\left\{ |S_Z|,\ M \right\}, \end{aligned} \tag{47a}$$

and the lower bound

$$\begin{aligned} F_S(Z) &\ge \min\left\{ (1-\epsilon)\lambda \cdot |S_Z|, (1-\epsilon)\lambda \cdot |S_Z| - \lambda(|S_Z| - M) \right\} \\ \frac{F_S(Z)}{\lambda} &\ge \min\left\{ (1-\epsilon)\cdot |S_Z|,\ M - \epsilon \cdot |S_Z| \right\}. \end{aligned} \tag{47b}$$

We denote $\epsilon = \frac{p_{\min}}{4}$ for notational simplicity. As an overview, for the two cases to be presented below, we apply the upper bound (47a) to the optimal set S^*, and the lower bound (47b) to our candidate sets which we are proving are competitive with S^*.

Case 1: $\mu_{[n]} > M$ and $n \ge \frac{36}{p_{\min}^2}$. Consider any arbitrary set $S \subseteq [n]$ that satisfies (cf. Line 4 of Algorithm 6):

$$M \le \sum_{i\in S} p_i < M + 1. \tag{48}$$

Such a set S always exists because each $p_i \le 1$. Furthermore, such a set in S can be found efficiently, by greedily adding items to the set in an arbitrary order one-by-one until condition (48) is satisfied. We denote by $\mathcal{E}$ the event that at most half of the Bernoulli random variables from S are 1. Formally, $\mathcal{E} \coloneqq \{|S_Z| \le M/2\}$. By the multiplicative Chernoff bound,

$$\mathbb{P}(\mathcal{E}) = \mathbb{P}\Big(|S_Z| \le \frac{M}{2}\Big) \overset{\text{(i)}}{\le} \Pr\Big(|S_Z| \le \frac{\mu_S}{2}\Big) \le e^{-\frac{\mu_S}{8}} \overset{\text{(ii)}}{\le} e^{-\frac{M}{8}}, \tag{49}$$

where steps (i) and (ii) are true by the construction of S in (48). We derive a lower bound on $F_S(Z)$ depending on $\mathcal{E}$. Conditional on $\mathcal{E}$, the penalty term is 0 and we have $F_S(Z) \ge 0$. We now consider the case conditional on $\overline{\mathcal{E}}$. By the assumption of $\mu_S < M + 1$ from (48), we have the deterministic relation $|S_Z| \le \frac{M+1}{p_{\min}}$. Applying the lower bound in (47b), conditional on $\overline{\mathcal{E}}$,

$$\begin{aligned}
F_S(Z) &\ge \lambda \cdot \min\left\{(1-\epsilon)\cdot\frac{M}{2},\ M - \epsilon\cdot|S_Z|\right\} \\
&\ge \lambda \cdot \min\left\{\frac{(1-\epsilon)M}{2},\ M - \frac{\epsilon(M+1)}{p_{\min}}\right\} \\
&= \lambda M \cdot \min\left\{\frac{1-\epsilon}{2},\ 1 - \frac{\epsilon(M+1)}{M\cdot p_{\min}}\right\} \\
&\overset{\text{(i)}}{=} \lambda M \cdot \frac{1-\epsilon}{2}
\end{aligned}$$

where step (i) holds because by the assumption $\epsilon \le \frac{p_{\min}}{4}$ and $M \ge 1$ (recall that $M \in \mathbb{N}_+$), we have $\frac{\epsilon}{p_{\min}} \le \frac{1}{4}$ and $\frac{M+1}{M} \le 2$. Therefore, we have $1 - \frac{\epsilon(M+1)}{M\cdot p_{\min}} \ge \frac{1}{2}$. Taking an expectation over Z, we then have

$$\begin{aligned}
U(S) &= \mathbb{E}[F_S(Z)] \\
&\ge 0 \cdot \Pr(\mathcal{E}) + \frac{(1-\epsilon)\lambda M}{2}\cdot\Pr(\overline{\mathcal{E}}) \\
&\overset{\text{(i)}}{\ge} \frac{(1-\epsilon)(1-e^{-\frac{M}{8}})}{2}\cdot\lambda M \\
&\overset{\text{(ii)}}{\ge} \frac{(1-\epsilon)(1-e^{-\frac{M}{8}})}{2}\cdot U(S^*),
\end{aligned}$$

where step (i) is true by (49), and step (ii) is true by applying (47a) to the optimal selection S^*. Since $\epsilon = \frac{p_{\min}}{4} \le \frac{1}{4}$ and $M \ge 1$ by assumption, this guarantees a constant-factor approximation of S to the optimal subset S^*.

Case 2: $\mu_{[n]} \le M$ and $n \ge \frac{36}{p_{\min}^2}$. In this case n is large enough to apply concentration bounds, so we downsample the set $[n]$ by a factor of two and discount the probability that $|S_Z|$ exceeds M. This differs from Case 1 in that we cannot compare our objective against $\lambda \cdot M$. In particular, we consider any arbitrary set $S \subseteq [n]$ that satisfies (cf. Line 6 of Algorithm 6):

$$\frac{\mu_{[n]}}{3} \le \mu_S \le \frac{\mu_{[n]}}{2} \le \frac{M}{2}. \tag{50}$$

We first show that such a set S always exists. Note that we have $\mu_{[n]} \ge np_{\min} \ge np_{\min}^2 \ge 36$ by the assumption of $n \ge \frac{36}{p_{\min}^2}$. Hence, $\frac{\mu_{[n]}}{2} - \frac{\mu_{[n]}}{3} \ge 1$. Since each $p_i \le 1$, a set S always exists. Moreover, it can be found efficiently, by greedily adding items one-by-one in any arbitrary order until condition (50) is satisfied. In what follows, we separately bound the values of $U(S)$ and $U(S^*)$. We use an intermediate quantity of the expectation of the random variable $|S_Z|$ truncated at $2\mu_S$, defined by

$$G(|S_Z|) \coloneqq \mathbb{E}\Big[|S_Z| \cdot \mathbb{1}\{|S_Z| \le 2\mu_S\}\Big].$$

Lower bound on $U(S)$. Due to the condition (50) that $\mu_S \le \frac{M}{2}$, we have

$$\begin{aligned}
G(|S_Z|) &\coloneqq \mathbb{E}\Big[|S_Z| \cdot \mathbb{1}\{|S_Z| \le 2\mu_S\}\Big] \\
&\le \mathbb{E}\left[|S_Z| \cdot \mathbb{1}\{|S_Z| \le M\}\right].
\end{aligned} \tag{51}$$

We claim the deterministic relation

$$\min\left\{|S_Z|,\ \frac{M-\epsilon|S_Z|}{(1-\epsilon)}\right\} \geq |S_Z|\cdot \mathbb{1}\{|S_Z|\leq M\}. \tag{52}$$

To see (52), we observe that when $|S_Z|\leq M$, the left-hand side has

$$\frac{1}{1-\epsilon}\left(M-\epsilon|S_Z|\right) \geq M \geq |S_Z|. \tag{53a}$$

When $|S_Z|>M$, the right-hand side is zero, and the left-hand side is nonnegative, because

$$M \geq 2\mu_S \geq 2p_{\min}|S| \geq 2p_{\min}|S_Z| \geq \epsilon|S_Z|. \tag{53b}$$

Plugging (52) to (51), we have

$$\begin{aligned}
G(|S_Z|) &\leq \mathbb{E}\min\left\{|S_Z|,\ \frac{M-\epsilon|S_Z|}{1-\epsilon}\right\} \\
&\leq \frac{1}{1-\epsilon}\mathbb{E}\min\left\{|S_Z|,\ \frac{M-\epsilon|S_Z|}{1-\epsilon}\right\} \\
&= \frac{1}{(1-\epsilon)^2}\mathbb{E}\Big[\min\left\{(1-\epsilon)\cdot|S_Z|,\ M-\epsilon\cdot|S_Z|\right\}\Big] \\
&\overset{\text{(i)}}{\leq} \frac{1}{(1-\epsilon)^2}\cdot\frac{U(S)}{\lambda},
\end{aligned} \tag{54}$$

where step (i) follows from (47b).

Upper bound on $U(S^*)$. We decompose the expectation of $|S_Z|$ as

$$\mathbb{E}|S_Z| = G(|S_Z|) + \mathbb{E}\left[|S_Z|\cdot\mathbb{1}\{|S_Z|>2\mu_S\}\right],$$

and hence

$$\begin{aligned}
G(|S_Z|) &= \mu_S - \mathbb{E}\Big[|S_Z|\cdot\mathbb{1}\{|S_Z|>2\mu_S\}\Big] \\
&\overset{\text{(i)}}{\geq} \mu_S - \mathbb{E}\Big[|S_Z|\cdot\mathbb{1}\{|S_Z|>M\}\Big] \\
&= \mu_S - \mathbb{E}\Big[M\cdot\mathbb{1}\{|S_Z|>M\} + (|S_Z|-M)\cdot\mathbb{1}\{|S_Z|>M\}\Big] \\
&= \mu_S - \underbrace{M\cdot\mathbb{P}\Big(|S_Z|>M\Big)}_{T_1} - \underbrace{\mathbb{E}\Big[(|S_Z|-M)_+\Big]}_{T_2},
\end{aligned} \tag{55}$$

where step (i) is true by the condition (50) that $\mu_S \leq \frac{M}{2}$. We now analyze the two terms T_1 and T_2 separately. We define δ such that $M=(1+\delta)\mu_S$, and we have $\delta\geq 1$ by the construction of S in (50).

For the term T_1, we apply the multiplicative Chernoff bound. We have

$$\mathbb{P}\Big(|S_Z|>M\Big) \leq \mathbb{P}\Big(|S_Z|>2\mu_S\Big) \leq e^{-\frac{\mu_S}{3}}.$$

Then

$$T_1 \leq 2\mu_S\cdot e^{-\frac{\mu_S}{3}} \overset{\text{(i)}}{\leq} \frac{e}{6}, \tag{56}$$

where it can verified that step (i) holds for any $\mu_S\in\mathbb{R}$.

For the term T_2, note that $T_2=\frac{V(S)}{\lambda}$, and by condition (50) we have $\mu_S\leq\frac{M}{2}\leq M$. Applying Lemma 8 with $k=\lceil\frac{1}{p_{\min}}\rceil$ yields

$$T_2 \leq \left\lceil\frac{1}{p_{\min}}\right\rceil\cdot\mathrm{Exp}\left(\frac{-2(M-\mu_S)^2}{|S|}\right)\cdot\left(1-e^{\frac{-4(M-\mu_S)}{|S|}\left\lceil\frac{1}{p_{\min}}\right\rceil}\right)^{-2}.$$

Using the relations $|S| \le \frac{\mu_S}{p_{\min}}$ and $M - \mu_S \ge \mu_S$, we have

$$\begin{aligned} T_2 &\le \left\lceil \frac{1}{p_{\min}} \right\rceil \cdot \operatorname{Exp}\left(\frac{-2\mu_S^2}{\mu_S / p_{\min}} \right) \cdot \left(1 - e^{\frac{-4\mu_S}{\mu_S / p_{\min}} \left\lceil \frac{1}{p_{\min}} \right\rceil} \right)^{-2} \\ &= \left\lceil \frac{1}{p_{\min}} \right\rceil \cdot \operatorname{Exp}\left(-2 p_{\min} \mu_S \right) \cdot \left(1 - e^{-4} \right)^{-2} \\ &\le \left\lceil \frac{1}{p_{\min}} \right\rceil \cdot \left(1 - e^{-4} \right)^{-2}. \end{aligned} \tag{57}$$

Plugging term T_1 from (56) and term T_2 from (57) back to (55) yields

$$G(|S_Z|) \ge \mu_S - \frac{6}{e} - 1.04 \cdot \left\lceil \frac{1}{p_{\min}} \right\rceil.$$

Recall from the construction of S in (50) that $\frac{\mu_{[n]}}{3} \le \mu_S \le \frac{\mu_{[n]}}{2}$. Furthermore, by the assumption that $n \ge \frac{36}{p_{\min}^2}$, we have

$$\mu_{[n]} \ge n p_{\min} \ge \frac{36}{p_{\min}} \ge \max\left\{ 36, 18 \left\lceil \frac{1}{p_{\min}} \right\rceil \right\}. \tag{58}$$

Hence, we have

$$G(|S_Z|) \ge \frac{\mu_{[n]}}{3} - \frac{\mu_{[n]}}{12} - \frac{\mu_{[\mu_S]}}{12} \ge \frac{\mu_{[n]}}{6}.$$

Applying inequality (47a) with the fact that $\mathbb{E}[|S_Z|] \le \mu_{[n]}$, we have

$$U(S^*) \le \lambda \cdot \mu_{[n]}$$

and hence

$$G(|S_Z|) \ge \frac{\mu_{[n]}}{6} \ge \frac{U(S^*)}{6\lambda}. \tag{59}$$

Finally, combining (54) and (59) yields

$$\frac{U(S)}{U(S^*)} \ge \frac{\lambda(1-\epsilon)^2 \cdot G(|S_Z|)}{6\lambda \cdot G(|S_Z|)} = \frac{(1-\epsilon)^2}{6},$$

yielding a constant-factor approximation with $\epsilon = \frac{p_{\min}}{4} \le \frac{1}{4}$.

Runtime. MEDIUMVALUEL$_1^+$ (Algorithm 6) begins by brute forcing over small sets, and there are $2^n \le 2^{36/p_{\min}^2} \le n^{36/p_{\min}^2}$ such sets. By Lemma 5, the objective value for each such set can be evaluated in polynomial time, and so the runtime in this case is $n^{3+36/p_{\min}^2}$.

For the other two cases (Line 3 and Line 5), the chosen set can be identified in $O(n)$. Therefore the overall runtime of MEDIUMVALUEL$_1^+$ is $n^{O(1/p_{\min}^2)}$. □

B.8.3 Proof of Lemma 5

Recall from (1) that the objective is computed as $U([n]) = R([n]) - \lambda \cdot V([n])$, with

$$R([n]) \coloneqq \sum_{i \in [n]} p_i x_i$$

$$V([n]) \coloneqq \mathbb{E}\, \rho\left(\sum_{i \in [n]} Z_i,\ M \right).$$

It is clear that computing the reward term R may be done in $O(n)$ operations. We now show that the penalty term V can be computed in $O(n^2)$ operations.

We start by rewriting the term V as:

$$V([n]) = \sum_{k=0}^{n} \mathbb{P}\Big(\sum_{i\in[n]} Z_i = k \Big) \cdot \rho(k, M) \tag{60}$$

For any integer $m \in \{0, 1, \ldots, n\}$, we define the $(m+1)$-dimensional vector $\{w^{(m)}\}_{k=0}^{m}$ by

$$w_k^{(m)} \coloneqq \mathbb{P}\Big(\sum_{i\in[m]} Z_i = k \Big).$$

Since we assume that the relevant values of ρ are known at the outset, it suffices to show that the probabilities involved in (60), or equivalently the $(n+1)$-dimensional vector $\{w_k^{(n)}\}_{k=0}^{n}$, can be computed in $O(n^2)$ operations.

We iteratively compute the vector of $\{w_k^{(m)}\}_{k=0}^{m}$ for $m \in \{0, 1, \ldots, n\}$. First, we observe that $w^{(0)} = 0$. Then we observe the iterative relation that for each $m \in [n]$ and $k \in \{0, \ldots, m\}$, we have

$$w_k^{(m)} = p_m \cdot w_{k-1}^{(m-1)} + (1 - p_m) \cdot w_k^{(m-1)}.$$

Hence, given the values of the m-dimensional vector $\{w_k^{(m-1)}\}_{k=0}^{m-1}$, computing each term $w_k^{(m)}$ takes c operations, where c is a universal constant. Hence, given the values of the m-dimensional vector $w_{\cdot}^{(m-1)}$, it takes $c(m+1)$ operations to compute the $(m+1)$-dimensional vector $w_{\cdot}^{(m)}$. Hence, the number of operations for computing the vector $w_{\cdot}^{(n)}$, by iteratively taking $m \in \{1, 2, \ldots, n\}$, is

$$c \sum_{m=1}^{n} (m+1) = O(n^2),$$

completing the proof. □

B.8.4 Proof of Lemma 6

We re-index the items $\{p_i, x_i\}_{i\in[n]}$ in decreasing order of the value x_i, such that $x_1 \geq \ldots \geq x_n$.

First, note that if $p_1 > \beta \cdot \mu_S$, then $T = \emptyset$ satisfies the lemma. Clearly for this T (30a) holds. Then because $R(S) \geq 0$ we also have

$$0 > \beta - \frac{p_i}{\mu_S} \geq \beta - \frac{1}{\mu_S} \geq \beta \left(1 - \frac{1}{\beta \cdot p_{\min}|S|}\right),$$

and so multiplying by $R(S)$ yields

$$R(T) = 0 > \beta \left(1 - \frac{1}{\beta \cdot p_{\min}|S|}\right) \cdot R(S),$$

satisfying (30b).

Otherwise we assume that $p_1 \leq \beta \cdot \mu_S$. We construct a set T be selecting as many items as possible in the decreasing order of the value x_i, subject to the constraint that (30a) is satisfied. Formally, we consider the set $T \coloneqq \{1, \ldots, t\}$, where

$$t \coloneqq \max \left\{ m \in [n] : \sum_{i=1}^{m} p_i \leq \beta \cdot \mu_S \right\}.$$

By the definition of t, the set T satisfies (30a). It remains to show that the set T also satisfies (30b).

If $\beta = 1$ then the resulting $T = S$ clearly suffices. Otherwise $\beta < 1$, and so we have $t < n$ (we assume that each item has strictly positive probability without loss of generality. By the definition of t, we have $\sum_{i=1}^{t+1} p_i > \beta \cdot \mu_S$. Equivalently,

$$\mu_T > \beta \cdot \mu_S - p_{t+1}. \tag{61}$$

In what follows, we use the following inequality that holds for any for $\{a_i\}_{i\in[n]}$ and $\{b_i\}_{i\in[n]}$ with $b_i \geq 0$:

$$\min_i \frac{a_i}{b_i} \leq \frac{\sum_i a_i}{\sum_i b_i} \leq \max_i \frac{a_i}{b_i}. \tag{62}$$

To see why this is true, note that for any $\{r_i\}_{i\in[n]}$ and $\{w_i\}_{i\in[n]}$, with $w_i \geq 0$ and $\sum_i w_i = 1$, we have

$$\min_i r_i \leq \sum_i w_i r_i \leq \max_i r_i.$$

We recover (62) by setting $r_i = \frac{a_i}{b_i}$ and $w_i = \frac{b_i}{\sum_i b_i}$.

Applying (62) yields

$$\frac{R(T)}{\mu_T} = \frac{\sum_{i\in T} p_i x_i}{\sum_{i\in T} p_i} \overset{\text{(i)}}{\geq} x_t \overset{\text{(ii)}}{\geq} \frac{\sum_{i\in S\setminus T} p_i x_i}{\sum_{i\in S\setminus T} p_i} = \frac{R(S\setminus T)}{\mu_S - \mu_T}, \tag{63}$$

where steps (i) and (ii) hold because the items are sorted in the decreasing order of x_i. Plugging $R(S) = R(T) + R(S\setminus T)$ into (63) and rearranging yields

$$\begin{aligned} R(T) &\geq \frac{\mu_T}{\mu_S} R(S) \\ &\overset{\text{(i)}}{\geq} \left(\beta - \frac{1}{\mu_S}\right) R(S) \\ &\overset{\text{(ii)}}{\geq} \left(\beta - \frac{1}{p_{\min}\cdot|S|}\right) R(S), \end{aligned}$$

where step (i) is true by (61), and step (ii) follows again from the fact that $\mu_S \geq p_{\min}|S|$. Hence, the set T satisfies (30b), completing the proof. □

B.8.5 Proof of Lemma 7

To begin, we partition S into "high," "bucketable," and "leftover" items according to their p_i so that $S = H \sqcup B \sqcup L$ in Algorithm 7.

Algorithm 7 PARTITION

Require: $S \in [n], p \in [0,1]^n$
Ensure: A partition $S = H \sqcup B \sqcup L$ with $B = D_1 \sqcup D_2 \sqcup D_3$

$H \leftarrow \{i \in S : p_i \geq \frac{1}{4}\}$
$L \leftarrow \{\}$
$D_1, D_2, D_3 \leftarrow \{\}$
for $\ell = 2, \ldots, \left\lceil \log_2(\frac{1}{p_{\min}}) \right\rceil - 1$ **do**
  $B^\ell \leftarrow \{i \in S : 2^{-(\ell+1)} \leq p_i < 2^{-\ell}\}$
  for $j = 0, \ldots, \left\lfloor \frac{|B^\ell|}{3} \right\rfloor - 1$ **do**
    $B_j^\ell \leftarrow \{b_{3j+1}^\ell, b_{3j+2}^\ell, b_{3j+3}^\ell\}$
    $D_1 \leftarrow D_1 \cup \{b_{3j+1}^\ell\}$
    $D_2 \leftarrow D_2 \cup \{b_{3j+2}^\ell\}$
    $D_3 \leftarrow D_3 \cup \{b_{3j+3}^\ell\}$
  $L \leftarrow L \cup \{b^\ell_{\left\lfloor 3\frac{|B^\ell|}{3}\right\rfloor+1}, \ldots, b^\ell_{|B^\ell|}\}$
return $S = H \sqcup B \sqcup L$ with $B = D_1 \sqcup D_2 \sqcup D_3$

Algorithm 7 first let $H = \{i \in S : p_i \geq \frac{1}{4}\}$, the high-probability items. Next consider the collection of buckets $B^\ell = \{i \in S \setminus H : 2^{-(\ell+1)} \leq p_i < 2^{-\ell}\}$. Note that the number of buckets is at most $\log_2(\frac{1}{p_{\min}})$. Form the contents of each B^ℓ into groups of three, $\{B_j^\ell\}_j$ (that is, the set $\left|B_j^\ell\right| = 3$ for

each j. If the number of items in B^ℓ is not divisible by 3, we leave them to L). Let $B = \cup_\ell \cup_j B_j^\ell$, and let L be the leftover $L \coloneqq S \setminus (H \cup B)$ which do not belong to groups of three.

Next note that $R(H) + R(B) + R(L) = R(S)$ and that $V(H), V(B), V(L) \le V(S)$. We handle the cases when each of these is large separately.

Case 1: $R(H) \ge \frac{R(S)}{3}$**.** If $|H| \le \frac{24}{p_{\min}}$ then the set H satisfies (32a). We now consider the case $|H| > \frac{24}{p_{\min}}$, and construct a set $T \subseteq H$ that satisfies (32b).

Applying Lemma 6 with $k = 6$ yields a set $T \subseteq H$ such that

$$\mu_T \le \frac{1}{6}\mu_H \tag{64a}$$

and

$$R(T) \overset{\text{(i)}}{\ge} \frac{1}{6} \cdot \left(1 - \frac{6}{p_{\min} \cdot |H|}\right) \cdot R(H) \overset{\text{(ii)}}{\ge} \frac{1}{8} R(H), \tag{64b}$$

where step (i) follows from Lemma 6 and step (ii) is true by the assumption that $|H| > \frac{24}{p_{\min}}$. By the definition of H, we have $p_i \ge 1/4$ for each $i \in H$, and hence

$$|T| \le 4\mu_T \overset{\text{(i)}}{\le} \frac{2}{3}\mu_H \le \frac{2}{3}\mu_S \overset{\text{(ii)}}{\le} M,$$

where step (i) is true by (64a), and step (ii) is true by the assumption that $\mu_S \le \frac{3}{2}M$. Hence, we have $V(T) = V(T') = 0$. By the rounding procedure, we have $R(T') = R(T)$. Therefore,

$$U(T') = U(T) = R(T) \overset{\text{(i)}}{\ge} \frac{1}{8} R(H) \overset{\text{(ii)}}{\ge} \frac{1}{24} R(S) \ge \frac{1}{24} U(S), \tag{65}$$

where step (i) is due to (64b) and step (ii) is true by the assumption of this case. Hence, the set T satisfies the condition (32b).

Case 2: $R(L) \ge \frac{R(S)}{3}$**.** Recall that the number of buckets is at most $\log_2(\frac{1}{p_{\min}})$. Since there are at most two elements in L from each bucket, the number of items in L is at most $|L| \le 2\log_2(\frac{1}{p_{\min}}) < \frac{2}{p_{\min}}$, satisfying condition (32a).

Case 3: $R(B) \ge \frac{R(S)}{3}$**.** Further partition B into three equal-sized sets $B = D_1 \sqcup D_2 \sqcup D_3$ by arbitrarily assigning each member of each bucket-group B_j^ℓ to a distinct D_ℓ. Without loss of generality, assume that D_1 has the maximum reward among these three sets, namely $R(D_1) \ge \max\{R(D_2), R(D_3)\}$, so that $R(D_1) \ge \frac{R(B)}{3} \ge \frac{R(S)}{9}$.

In what follows, we first show that the set D_1 satisfies (32b) under the assumption

$$V(D_1') \le V(S). \tag{66}$$

Then we show that assumption (66) always holds.

Proving (32b) for set D_1. For the reward term, we have

$$R(D_1') = R(D_1) \ge \frac{1}{9} R(S).$$

For the penalty term, recall that we assume $\lambda \cdot V(S) \le \frac{1}{15} R(S)$. Combining the reward term and the penalty term, we have

$$\begin{aligned} U(D_1') &= R(D_1') - \lambda \cdot V(D_1') \\ &\overset{\text{(i)}}{\ge} \frac{1}{9} R(S) - \lambda \cdot V(S) \\ &\ge \frac{1}{9} R(S) - \frac{1}{15} R(S) \\ &\ge \frac{1}{24} U(S), \end{aligned}$$

where step (i) uses assumption (66). Hence, the set D_1' satisfies condition (32b). It remains to prove assumption (66).

Proving (66). For any sets S_1 and S_2, we say that S_1 stochastically dominates S_2, if the random variable $\sum_{i\in S_1} Z_i$ stochastically dominates the random variable $\sum_{i\in S_2} Z_i$. Namely, for any $t \in \mathbb{R}$, we have

$$\mathbb{P}\Big(\sum_{i\in S_1} Z_i \geq t\Big) \geq \mathbb{P}\Big(\sum_{i\in S_2} Z_i \geq t\Big)$$

Since the one-sided loss L_1^+ is nondecreasing, it can be verified that if S_1 stochastically dominates S_2, then $V(S_1) \geq V(S_2)$.

By construction we have $B \subseteq S$, and hence S stochastically dominates B. If B stochastically dominates D_1', then we have

$$V(D_1') \leq V(B) \leq V(S),$$

proving (66). It remains to prove that B stochastically dominates D_1'.

For each bucket group $B_z = B_j^\ell$, let $B_z = \{p_1, p_2, p_3\}$ with $p_1 \in D_1$. Then let the associated random variables be

$$X_z \coloneqq \text{Ber}(q_1) \qquad \text{and} \qquad Y_z \coloneqq \text{Ber}(p_1) + \text{Ber}(p_2) + \text{Ber}(p_3),$$

where q_i is obtained by rounding p_i up to the nearest power of two. Note that $\sum_{i\in D_1'} Z_i' = \sum_z X_z$, and $\sum_{i\in B} Z_i = \sum_z Y_z$. Moreover, $\{X_z\}_z$ are independent, and $\{Y_z\}_z$ are independent. It suffices to show the stochastic dominance of Y_z over X_z for each bucket group b_z, and then the stochastic dominance of B over D_1' follows.

To show the stochastic dominance of Y_z over X_z, we consider the probabilities

$$\begin{aligned}\mathbb{P}(X_z = 0) &= 1 - q_1\\ \mathbb{P}(Y_z = 0) &= (1-p_1)(1-p_2)(1-p_3),\end{aligned}$$

and show that $\mathbb{P}(X_z = 0) \geq \mathbb{P}(Y_z = 0)$. By the construction of each bucket group B_ℓ, we have $p_1, p_2, p_3 \in [2^{-(l+1)}, 2^{-l})$ and hence $q_1 = 2^{-l}$. Consequently, we have $p_1, p_2, p_3 \geq \frac{q_1}{2}$. We have

$$\mathbb{P}(X_z = 0) = 1 - q_1 \overset{\text{(i)}}{\geq} \left(1 - \frac{q_1}{2}\right)^3 \geq (1-p_1)(1-p_2)(1-p_3) = \mathbb{P}(Y_z = 0).$$

where it can be verified that step (i) holds for every $q_1 \in [0, \frac{1}{4}]$. Hence, Y_z stochastically dominates X_z, completing the proof of (66). □

B.8.6 Proof of Lemma 8

For notational simplicity, we assume $\lambda = 1$ without loss of generality, and denote the random variable $T \coloneqq \sum_{i\in S} Z_i$. We write the penalty term $V(S)$ as

$$\begin{aligned}V(S) &= \mathbb{E}\,(T - M)_+\\ &= \sum_{i=1}^{\infty} i \cdot \mathbb{P}\Big(T = M + i\Big),\end{aligned} \tag{67}$$

We consider the probability that the value of T lies in each interval $(M + ik, M + (i+1)k]$, for each integer $i \geq 0$. We have

$$\begin{aligned}V(S) &\leq \sum_{i=0}^{\infty} (i+1)k \cdot \mathbb{P}\Big(M + ik < T \leq M + (i+1)k\Big)\\ &\leq k \cdot \sum_{i=0}^{\infty} (i+1) \cdot \mathbb{P}\Big(T > M + ik\Big).\end{aligned}$$

We bound each term $\mathbb{P}\Big(T > M + ik\Big)$ by Hoeffding's inequality. We have $\mathbb{E}[T] = \mu_S \leq M$ by assumption. Hence, by Hoeffding's inequality,

$$\begin{aligned}
\mathbb{P}\Big(T > M + ik\Big) &\leq \text{Exp}\left(-\frac{2(M + ik - \mu_S)^2}{|S|}\right) \\
&= \text{Exp}\left(-\frac{2(M - \mu_S)^2}{|S|}\right) \cdot \text{Exp}\left(-\frac{4(M - \mu_S)ik + 2(ik)^2}{|S|}\right) \\
&\leq \text{Exp}\left(-\frac{2(M - \mu_S)^2}{|S|}\right) \cdot \text{Exp}\left(-\frac{4(M - \mu_S)ik}{|S|}\right). \qquad (68)
\end{aligned}$$

Plugging (68) into (67) yields

$$\begin{aligned}
V(S) &\leq k \cdot \text{Exp}\left(-\frac{2(M - \mu_S)^2}{|S|}\right) \cdot \sum_{i=0}^{\infty}(i+1) \cdot \text{Exp}\left(-\frac{4(M - \mu_S)ik}{|S|}\right) \\
&\leq k \cdot \text{Exp}\left(-\frac{2(M - \mu_S)^2}{|S|}\right) \cdot \sum_{i=0}^{\infty}(i+1) \cdot \left(e^{-\frac{4(M-\mu_S)k}{|S|}}\right)^i \\
&\overset{\text{(i)}}{=} k \cdot \text{Exp}\left(-\frac{2(M - \mu_S)^2}{|S|}\right) \cdot \left(1 - e^{\frac{-4(M-\mu_S)k}{|S|}}\right)^{-2},
\end{aligned}$$

where step (i) uses the fact that for any $0 < x < 1$, we have

$$\sum_{i=0}^{\infty}(i+1)x^i = \sum_{t=0}^{\infty}\sum_{i=t}^{\infty}x^i = \sum_{i=0}^{\infty}\frac{x^t}{1-x} = \frac{1}{1-x}\sum_{i=0}^{\infty}x^t = \frac{1}{(1-x)^2}.$$

□

B.9 Proof of Theorem 5

We fix any arbitrary problem instance (x, p, λ, M) for the L_1 loss, and problem instance (x', p, λ', M) for the L_1^+ loss, with

$$\begin{aligned}
x'_i &:= x_i - \lambda \\
\lambda' &:= 2\lambda.
\end{aligned}$$

We fix any arbitrary $S \subseteq [n]$, and demonstrate the desired equality

$$U_{L_1}(S) = U_{L_1^+}(S') - \lambda \cdot M. \qquad (69)$$

by induction on the number of elements in S. First, we consider $|S| = 0$, or equivalently $S = \emptyset$. Then it can be verified that

$$\begin{aligned}
U_{L_1}(S) &= -\lambda M \\
U_{L_1^+}(S') &= 0,
\end{aligned}$$

satisfying (69).

Next suppose that (69) holds for all set S with $|S| \leq k$. We consider the marginal change to the objective when adding any item $j \notin S$ to the set S. Let $S_j =$ for some S with $|S| < j$. The marginal change to the objective with the L_1 loss is

$$\begin{aligned}
U_{L_1}(S \cup \{j\}) - U_{L_1}(S) &= p_j x_j + \lambda \cdot \mathbb{E}_{Z_{S\cup\{j\}}}\left[\left|\sum_{i\in S} Z_i + Z_j - M\right| - \left|\sum_{i\in S} Z_i - M\right|\right] \\
&= p_j x_j + \lambda \cdot p_j \cdot \mathbb{E}_{Z_S}\underbrace{\left[\left|\sum_{i\in S} Z_i + 1 - M\right| - \left|\sum_{i\in S} Z_i - M\right|\right]}_{T} \qquad (70)
\end{aligned}$$

Note that the term T satisfies

$$T = \begin{cases} 1 & \text{if } \sum_{i \in S} Z_i \geq M \\ -1 & \text{if } \sum_{i \in S} Z_i < M. \end{cases} \tag{71}$$

Using the fact (71) in (70), we have

$$\begin{aligned} U_{L_1}(S \cup \{j\}) - U_{L_1}(S) &= p_j x_j + \lambda \cdot p_j \left[\mathbb{P}\Big(\sum_{i \in S} Z_i \geq M \Big) - \mathbb{P}\Big(\sum_{i \in S} Z_i < M \Big) \right] \\ &= p_j x_j + \lambda \cdot p_j \left[2 \cdot \mathbb{P}\Big(\sum_{i \in S} Z_i \geq M \Big) - 1 \right] \\ &= p_j (x_j - \lambda) + 2\lambda p_j \cdot \mathbb{P}\Big(\sum_{i \in S} Z_i \geq M \Big) \\ &= p_j x'_j + \lambda' p_j \cdot \mathbb{P}\Big(\sum_{i \in S} Z_i \geq M \Big). \end{aligned} \tag{72}$$

Using a similar analysis, the marginal change to the objective with the L_1^+ loss is

$$\begin{aligned} U_{L_1^+}(S \cup \{j\}) - U_{L_1^+}(S) &= p_j x'_j + \lambda' \cdot \mathbb{E}_{Z_{S \cup \{j\}}} \left[\Big(\sum_{i \in S} Z_i + Z_j - M \Big)_+ - \Big(\sum_{i \in S} Z_i - M \Big)_+ \right] \\ &= p_j x'_j + \lambda' p_j \cdot \mathbb{P}\Big(\sum_{i \in S} Z_i \geq M \Big). \end{aligned} \tag{73}$$

Combining (72) and (73) demonstrates that the marginal change is equal for the L_1 and L_1^+ losses, under their respective instances. Therefore, applying the induction hypothesis that (69) holds for all S with $|S| \leq k$ completes the induction step. □