Dependent randomized rounding for clustering and partition systems with knapsack constraints

David G. Harris
U. Maryland

Thomas Pensyl
Bandwidth, Inc.

Aravind Srinivasan
U. Maryland

Khoa Trinh
Google

Abstract

Clustering problems are fundamental to unsupervised learning. There is an increased emphasis on *fairness* in machine learning and AI; one representative notion of fairness is that no single demographic group should be over-represented among the cluster-centers. This, and much more general clustering problems, can be formulated with "knapsack" and "partition" constraints. We develop new randomized algorithms targeting such problems, and study two in particular: multi-knapsack median and multi-knapsack center. Our rounding algorithms give new approximation and pseudo-approximation algorithms for these problems. One key technical tool we develop and use, which may be of independent interest, is a new tail bound analogous to Feige (2006) for sums of random variables with unbounded variances. Such bounds are very useful in inferring properties of large networks using few samples.

1 Introduction

Clustering is a classical technique in unsupervised learning, and our goal is to systematically study clustering with knapsack constraints, particularly in light of fairness. To explain the random processes we study in this paper, let us consider a data-clustering problem, with a set $\mathcal{F}$ of potential cluster-centers, a dataset $\mathcal{C}$, a symmetric distance-metric d on $\mathcal{F} \cup \mathcal{C}$, and a non-negative $m \times |\mathcal{F}|$ budget matrix M that corresponds to m normalized cost functions on $\mathcal{F}$. Our goal is to choose a set $\mathcal{S} \subseteq \mathcal{F}$ of cluster-centers so that $\sum_{i \in \mathcal{S}} M_{k,i} \leq 1$ for all k and such that the distances $d(j, \mathcal{S})$, for $j \in \mathcal{C}$, are "small". (The normalization with all ones in the right-hand side is without loss of generality.) The particular metric used to boil down the values $d(j, \mathcal{S})$ into an objective function is problem-specific. The constraints implied by the rows of the budget matrix M are referred to as *knapsack constraints*; the case $m > 1$ is often referred to as *multi-knapsack. We typically view m as "small", e.g., as a constant.*

We develop two key probabilistic techniques for our algorithms, both of which are quite general and may be of independent interest: (i) a new "Samuels-Feige" type of concentration inequality, which applies to *unbounded* random variables; and (ii) a new dependent randomized-rounding approach, which is called *Knapsack-Partition Rounding (KPR)* algorithm.

As a representative example, which is one of the main motivations behind this work, consider the following scenario. The sets $\mathcal{F}$ and $\mathcal{C}$ represent a population, with $\mathcal{F}$ possibly equaling $\mathcal{C}$ (or not); we want to partition $\mathcal{C}$ into a small number of clusters, with each cluster having some element of $\mathcal{F}$ as its cluster center or representative. The clusters need to have small radii, as in classical clustering. In addition, there is a collection of m given demographic groups $A_1, \ldots, A_m$, which are viewed as subsets of $\mathcal{F}$. We do not wish to disproportionately select our centers $\mathcal{S}$ from any demographic group. Thus, for each $k = 1, \ldots, m$, we have a constraint $|\mathcal{S} \cap A_k| \leq t_k$ where t_k is some target which is not much larger than the "fair" proportion of A_k compared to the general population. These can be represented as knapsack constraints; due to the normalization we use, the weight functions are given by $M_{k,i} = 1/t_k$ if $i \in A_k$ and $M_{k,i} = 0$ otherwise.

In clustering problems as ours, a common solution strategy is to first solve a related linear program (LP) depending on the chosen objective function, leading to a fractional configuration $y \in [0,1]^{\mathcal{F}}$ over $\mathcal{F}$. Here, the LP suggests the fractional extent y_i for $i \in \mathcal{F}$ to be chosen as a center. Based on this solution y, we then partition $\mathcal{F}$ into groups so as to select exactly one cen-

Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

ter in each group. Finally, we use some randomized-rounding algorithm to convert the fractional solution y into an integral solution $Y \in \{0,1\}^{\mathcal{F}}$ (representing the chosen solution $\mathcal{S}$), while ensuring that the knapsack constraints are satisfied and each group gets exactly one selected center. The vector Y should also have other probabilistic properties similar to the vector y, for example satisfying $\mathbf{E}[Y] = y$ coordinatewise. This rounding process often goes by the name "dependent rounding," since the entries of Y will not necessarily be independent; see, e.g., [Byrka et al., 2015, Byrka et al., 2010, Charikar and Li, 2012].

This paper focuses on the randomized-rounding part of this solution strategy. We develop a general rounding method for a fractional vector y in the presence of knapsack constraints plus a single partition constraint. Suppose we have a partition $G_1, \ldots, G_r$ of a ground set $\{1, \ldots, n\}$; we refer to the sets G_i as *blocks* of the partition. Our original fractional vector y satisfies $My \leq \vec{1}$ and $\sum_{j \in G_i} y_j = 1$ for each block G_i; we wish to find an integral vector Y which also satisfies these constraints. For the clustering problems, the block G_j is the set of potential cluster-centers "close to" item $j \in \mathcal{C}$.

These clustering problems pose a challenge for dependent rounding because they are fundamentally *nonlinear*: the distance from any point j to its closest center depends on the *joint* behavior of the selected centers. If *none* of the centers in a group are selected, then any nearby elements $j \in \mathcal{C}$ may get large values $d(j, \mathcal{S})$. Consequently, our rounding process must ensure strong near-independence properties.

Partition systems have an obvious rounding method which we refer to as *independent selection*: namely, exactly one item is chosen from each block G_i wherein j is chosen with probability y_j. This completely ignores the knapsack constraints. Our rounding strategy seeks to mimic the probabilistic guarantees of independent selection, while also preserving the knapsack constraints. Unfortunately, it is impossible to *exactly* satisfy these goals simultaneously. Nevertheless, we achieve significantly stronger guarantees in both of these dimensions compared to previous rounding algorithms. We summarize these next, and then discuss our new clustering results.

1.1 Additive pseudo-approximation

Knapsack constraints can be intractable to satisfy exactly, so one common strategy is *pseudo-approximation*, that is, finding a solution which only *approximately* satisfies the knapsack constraints. This should be distinguished from a true approximation algorithm, which finds a feasible solution whose objective function is within some constant factor of the optimal one. Many previous algorithms (e.g., [Byrka et al., 2015]) have focused on what we refer to as *ϵ-multiplicative pseudo-solutions*: namely that $\sum_{i \in \mathcal{S}} M_{k,i} \leq 1 + \epsilon$ for each k. One way in which our rounding algorithm deals with the tradeoffs between independence and knapsack constraints is that, instead of getting a solution vector Y which is *fully integral*, that is, $Y \in \{0,1\}^n$, it only achieves a vector $\tilde{Y}$ which is "mostly" integral, that is, $\tilde{Y} \in [0,1]^n$ has only a limited number q of fractional entries. Crucially, the value q is *constant* (for a desired approximation ratio).

This naturally leads to an alternative form of knapsack pseudo-approximation that we refer to as an *additive pseudo-solution* or as an *additive pseudo-approximation*. We define this formally as follows:

Definition 1.1 (additive pseudo-approximation) *For a knapsack constraint (vector of weights w with capacity 1), a set $\mathcal{S}$ is a q-additive pseudo-solution if it has the form $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1$, where $\sum_{i \in \mathcal{S}_0} w_i \leq 1$ and $|\mathcal{S}_1| \leq q$. (Equivalently, $\mathcal{S}$ satisfies the budget constraint after removing its q highest-weight items.)*

For a multi-knapsack constraint $M = M_1, \ldots, M_m$, we say that $\mathcal{S}$ is a q-additive pseudo-solution *for M if it is a q-additive solution for each of the m knapsack constraints $M_1, \ldots, M_m$ separately. That is, for each $k = 1, \ldots, m$, we have $\mathcal{S} = \mathcal{S}_0^{(k)} \cup \mathcal{S}_1^{(k)}$ wherein $\sum_{i \in \mathcal{S}_0^{(k)}} M_{k,i} \leq 1$ and $|\mathcal{S}_1^{(k)}| \leq q$.*

Such additive pseudo-solutions are naturally connected to the *method of alteration* in the probabilistic method, where we delete/alter some items in a random structure to establish a desired property [Alon and Spencer, 2016]. We naturally view $\mathcal{S}_1$ as a discarded set whose elements will be processed in an application-specific manner. Additive pseudo-approximations have appeared implicitly in prior algorithms, e.g., [Li and Svensson, 2016, Krishnaswamy et al., 2011]. We can summarize some advantages of their advantages here, from both practical and technical points of view.

First, additive pseudo-approximation can be a useful tool to obtain multiplicative pseudo-approximations or even true approximations. As one example, if we have a q-additive pseudo-solution with q constant, we can modify it by handling these q additional items in some problem-specific way. Since q is constant, this may incur only a small overhead in the cost or computational complexity. This strategy was used in the k-median approximation algorithm of [Li and Svensson, 2016], and we use it here for our (true) approximation algorithm for knapsack center. These problems are described in Section 1.3.

As another example, there is a common strategy for

getting multiplicative pseudo-approximation to knapsack problems by "guessing" – exhaustively enumerating – the set of "big" items (items with $M_i > \epsilon/q$) which appear in an optimal solution. In the residual problem, each item has small weight. Thus, a q-additive pseudo-solution for the residual problem yields an ϵ-multiplicative solution to the original problem. This can be more efficient than generating the multiplicative pseudo-solution directly. The reverse direction does not hold, in general; there does not seem to be any way to go from multiplicative to additive pseudo-solutions.

To see a practical advantage of additive pseudo-approximation, consider our motivating example concerning fair representation for a number of demographic groups. In this setting, a q-additive pseudo-solution $\mathcal{S}$ will lead to a relatively modest violation of the fairness constraint, namely, it will have only $|\mathcal{S} \cap A_k| \leq t_k + q$ for each population A_k and associated target value t_k. By contrast, an ϵ-multiplicative pseudo-solution $\mathcal{S}$ will give a substantially larger violation, namely $|\mathcal{S} \cap A_k| \leq t_k(1+\epsilon)$.

1.2 Dependent rounding and independence

In the most straightforward form of dependent rounding, we have a fractional solution $x \in [0,1]^n$ that we wish to round to an integral vector $X \in \{0,1\}^n$ such that $\mathbf{E}[X] = x$ and the cardinality constraint $\sum_i X_i = \sum_i x_i$ holds with probability one. For example, [Charikar and Li, 2012] applied dependent rounding as part of their 3.25-approximation algorithm for the k-median problem. The cardinality constraint can easily be replaced by a single knapsack constraint [Ageev and Sviridenko, 2004, Srinivasan, 2001]. Over the last two decades, increasingly-sophisticated dependent-rounding techniques have been used for optimization problems over various types of polytopes; see, e.g., [Ageev and Sviridenko, 2004, Srinivasan, 2001, Gandhi et al., 2006, Călinescu et al., 2011, Chekuri et al., 2011, Bansal, 2019].

Our new rounding algorithm generalizes this setting in two distinct ways: it allows multiple knapsack constraints, and it allows a partition matroid constraint. Given some fractional vector $y \in [0,1]^n$ which satisfies these constraints, our overarching question is: *how well can we approximate independence while preserving the knapsack and partition constraints?*

The standard dependent-rounding algorithm satisfies a limited but important form of negative correlation, namely the *negative cylinder property* [Chekuri et al., 2010, Byrka et al., 2015, Srinivasan, 2001, Gandhi et al., 2006]: for any set $S \subseteq \{1,\dots,n\}$ the rounded variables X_i satisfy the conditions

$$\mathbf{E}\Big[\prod_{i\in S} X_i\Big] \leq \prod_{i\in S} x_i, \tag{1}$$

$$\mathbf{E}\Big[\prod_{i\in S}(1-X_i)\Big] \leq \prod_{i\in S}(1-x_i). \tag{2}$$

For our clustering results, we will need more general forms of negative correlation. To provide intuition, let us compare how our algorithm works for the standard dependent-rounding scenario, with a single knapsack constraint and no partition constraint. (We emphasize that our algorithm can handle much more general scenarios.) Ideally, we would like for arbitrary disjoint sets $S, T \subseteq \{1,\dots,n\}$ to satisfy a similar "near-independence" property:

$$\mathbf{E}\Big[\prod_{i\in S} X_i \prod_{i\in T}(1-X_i)\Big] \approx \prod_{i\in S} x_i \prod_{i\in T}(1-x_i). \tag{3}$$

For example, [Byrka et al., 2015] showed this type of property for running dependent rounding with a random permutation of the input vector. Note that Eq. (3) cannot be preserved exactly in an integral solution; for example, if $x_1 = x_2 = 1/2$, then any integral solution must either satisfy $\mathbf{E}[X_1(1-X_2)] \geq 1/2$ or $\mathbf{E}[(1-X_1)X_2] \geq 1/2$. This is part of the reason why general negative correlation is much more challenging than the negative cylinder property.

To overcome this fundamental barrier, our rounding algorithm terminates with a vector which has a small number t of fractional entries. The precise sense in which we mimic independent rounding is somewhat technical, but, as one example, we get

$$\mathbf{E}[\prod_{i\in S} X_i \prod_{i\in T}(1-X_i)] \leq O(1/t) + \prod_{i\in S} x_i \prod_{i\in T}(1-x_i). \tag{4}$$

We emphasize that Eq. (4) is only a simplified form of our rounding results. We will need, and develop, much more general bounds; in particular, we will handle cases where $S \cup T$ has a large size but only a few elements are "significant." Also, note that the usual dependent-rounding algorithms ([Srinivasan, 2001] and its relatives) do not satisfy negative-correlation properties such as negative cylinder for $m > 1$.

Let us briefly compare these near-independence properties with other dependent rounding algorithms. The first main genre of such algorithms is based on variants of the Lovász Local Lemma (e.g., [Harris and Srinivasan, 2019,

Leighton et al., 2001, Srinivasan, 2006]). These algorithms have very good independence properties, but also lead to knapsack violations on the order of the "standard deviation."

A second genre of algorithm is based, like ours, on Brownian motion in the constraint polytope. These algorithms are often targeted to discrepancy minimization, see e.g., [Beck and Fiala, 1981, Karp et al., 1987, Bansal and Nagarajan, 2016, Bansal, 2019], where the central goal is to show concentration bounds on linear functions of the variables. This can be regarded as a special case of negative correlation, which is closely related to pairwise correlation (covariance). Our algorithm gets tighter bounds and finer negative-correlation properties, including correlations among many variables, by taking advantage of the special properties of the knapsack-partition constraints.

1.3 Clustering results

In describing our clustering results, it is convenient to use the language of classical facility location. We refer to $\mathcal{C}$ as a set of *clients*, $\mathcal{F}$ as a set of potential *facilities* or *centers*, and we say that $i \in \mathcal{F}$ is *open* if i is placed into the solution set $\mathcal{S}$. We thus interpret the clustering problem as opening a small number of facilities. For a set $\mathcal{S} \subseteq \mathcal{F}$ of open facilities, the distance $d(j, \mathcal{S})$ can be interpreted as the connection cost of client j.

We study two clustering problems in particular: the *knapsack-median* and *knapsack-center* problems. In the knapsack median problem, we seek to minimize the *total* connection cost $\sum_j d(j, \mathcal{S})$ subject to m knapsack constraints. The knapsack-center problem is the same except that the objective function is to minimize $\max_{j \in \mathcal{C}} d(j, \mathcal{S})$ instead of the sum. We will briefly describe our algorithms and results for these problems here.

Knapsack median was first studied by [Krishnaswamy et al., 2011], which obtained an additive pseudo-approximation with an approximation factor of 16. The current best true approximation factor is 7.08 for $m = 1$, due to [Krishnaswamy et al., 2018]. The special case when all facilities in $\mathcal{F}$ have unit weight and $m = 1$, known as the classical k-median problem, can be approximated to within a factor of $2.675 + \epsilon$ [Byrka et al., 2015].

Our algorithm for this problem is inspired by an approximation algorithm of [Li and Svensson, 2016] for k-median. The idea is to solve a relaxation called a "bi-point solution", which is a linear relaxation that is a convex combination of two solutions. They then use dependent rounding, wherein we open either a facility from the first (feasible) solution, or the set of all facilities nearby in the second solution. This gives a q-additive pseudo-solution for some constant q. The k-median algorithm of [Li and Svensson, 2016] has an additional postprocessing step to correct it to a true approximation.

Our algorithm uses KPR for the dependent rounding step, which allows it to work with knapsack constraints instead of just cardinality constraints. The post-processing step does not seem to work for knapsack median, so the algorithm yields an additive pseudo-approximation instead of a true solution. We can correct the q-additive pseduo-solution to a multiplicative pseudo-solution by guessing the big facilities. Our KPR rounding algorithm gives the following results for additive and multiplicative pseudo-approximations:

Theorem 1.2 *There is a polynomial-time algorithm for knapsack median with $m = 1$ which generates, for any $\gamma \in (0, 1)$, an $O(\frac{1}{\gamma})$-additive pseudo-solution $\mathcal{S}$ with $cost(\mathcal{S}) \leq (1 + \sqrt{3} + \gamma)OPT \leq 2.733 \times OPT$.*

There is an algorithm running in time $n^{O(\epsilon^{-1}\gamma^{-1})}$ for knapsack median with $m = 1$ which generates, for any $\gamma, \epsilon > 0$, an ϵ-multiplicative pseudo-solution $\mathcal{S}$ with $cost(\mathcal{S}) \leq (1 + \sqrt{3} + \gamma)OPT$.

We also consider multi-knapsack median, where it is NP-hard to obtain a true approximation. Here, we adapt an algorithm of [Charikar and Li, 2012], which is based on rounding an LP solution. Based on the LP solution, this algorithm bundles the facilities together into pairs and ensures that at least one facility from each pair is opened. It then uses dependent rounding to ensure that, in addition to this hard constraint, each facility is opened with the proper probability. We apply KPR for the dependent rounding step to get a pseudo-approximation algorithm, which can also be leveraged into a multiplicative pseudo-approximation. We summarize the results here as follows:

Theorem 1.3 *For any parameter $\gamma \in (0, 1)$, there is a polynomial-time algorithm for multi-knapsack median with $m \geq 1$ constraints to get an $O(\frac{m}{\sqrt{\gamma}})$-additive pseudo-solution $\mathcal{S}$ with $cost(\mathcal{S}) \leq (3.25 + \gamma)OPT$.*

For any parameters $\gamma, \epsilon \in (0, 1)$, there is an algorithm in $n^{\tilde{O}(\frac{m^2}{\epsilon\sqrt{\gamma}})}$ time for multi-knapsack median to get an ϵ-multiplicative pseudo-solution $\mathcal{S}$ with $cost(\mathcal{S}) \leq (3.25 + \gamma)OPT$.

By contrast, if we directly used independent selection in the Charikar-Li algorithm, it would require $n^{\tilde{O}(m/\epsilon^2)}$ runtime to obtain an ϵ-multiplicative approximation. This illustrates how additive pseudo-approximation can be useful for efficient multiplicative pseudo-approximations; in particular, this gives a better dependence on the parameter ϵ (though a worse

dependence upon m, which we view as a small constant).

The knapsack center problem with $m = 1$ was first studied by [Hochbaum and Shmoys, 1986], under the name "weighted k-center". They gave a 3-approximation algorithm and proved that this is best possible unless P = NP; see also [Khuller et al., 2000]. For this case, our approximation algorithms find a feasible solution such that (1) all clients have distance at most $3 \times$ OPT to an open facility, and (2) all clients have expected connection cost at most $(1 + 2/e) \times$ OPT $\approx 1.74 \times$ OPT. Note that [Harris et al., 2018] showed that the constant factor $1 + 2/e$ cannot be improved unless P = NP, even in the k-supplier setting.

Our algorithm is based on a preprocessing step, which identifies and guesses facilities which serve a large fraction of the clients. After grouping the remaining clients, we then use our dependent rounding algorithm to select which facilities to open. This generates a q-additive pseudo-solution for $q = O(1)$ with the desired connection costs. We then modify this to get a true solution. Since each modified facility serves only a small number of clients, this has minimal effect on the connections costs. We summarize this result as follows:

Theorem 1.4 *For any $\gamma \in (0, 1)$, there is an algorithm running in $n^{\tilde{O}(1/\gamma)}$ time for the knapsack center problem with $m = 1$ which returns a feasible solution $\mathcal{S}$ such that every client $j \in \mathcal{C}$ has*

$$\mathbf{E}[d(j, \mathcal{S})] \leq (1 + 2/e + \gamma) \times OPT,$$

and

$$d(j, \mathcal{S}) \leq 3 \times OPT \text{ with probability one.}$$

More recently, [Chen et al., 2016] considered the case $m > 1$. They showed that it is intractable to obtain a true constant-factor approximation, and gave a multiplicative pseudo-approximation algorithm with approximation ratio 3. We obtain the following additive pseudo-approximations for this setting:

Theorem 1.5 (Simplified) *Let $\epsilon, \gamma \in (0, 1)$ and consider a knapsack-center problem with m constraints. We describe three algorithms to generate different types of pseudo-solutions $\mathcal{S}$ such that every client $j \in \mathcal{C}$ has $d(j, \mathcal{S}) \leq 3 \times OPT$ with probability one and $\mathbf{E}[d(j, \mathcal{S})] \leq (1 + 2/e + \gamma) \times OPT$.*

1. *There is a polynomial-time algorithm to get an $\tilde{O}(m/\sqrt{\gamma})$-additive pseudo-solution.*
2. *There is an algorithm with run-time $n^{\tilde{O}(m^2/\gamma)}$ to get an $\tilde{O}(\sqrt{m})$-additive pseudo-solution.*
3. *There is an algorithm with run-time $n^{\tilde{O}(m^{3/2}/\epsilon + m^2/\gamma)}$ for an ϵ-multiplicative pseudo-solution.*

The algorithms of [Hochbaum and Shmoys, 1986] or [Chen et al., 2016] may return a solution where nearly all the clients get service at distance $3 \times$ OPT. Our algorithm thus gives finer guarantees: every client gets service within $3 \times$ OPT with probability 1, but the average cost of service is much better. This can be helpful in *flexible* facility location, such as a streaming-service provider periodically reshuffling its service locations. It can also be interpreted as a type of fairness in clustering, where the fairness is in terms of individual users instead of demographic groups.

1.4 Notation

For any integer $t \geq 1$, we write $[t] = \{1, \dots, t\}$. For a set $X \subseteq [n]$ and an n-dimensional vector y, we write $y(X) = \sum_{i \in X} y_i$. For a distance function $d(x, y)$ and a set Y, we write $d(x, Y) = \min_{y \in Y} d(x, y)$. We use the $\tilde{O}()$ notation, so that $\tilde{O}(x) = O(x \cdot \text{polylog}(x)) = O(x(\log x)^c)$ for some constant $c > 0$.

2 Knapsack-partition systems and rounding

Our dependent rounding algorithm, which we call *Knapsack-Partition Rounding (KPR)*, is used for a class of problems we refer to as *knapsack-partition constraints*. Here, we are given a partition $\mathcal{G}$ over ground-set $[n]$ as well as a real-valued $m \times n$ matrix M. We refer to the sets $G \in \mathcal{G}$ as *blocks*. Finally, we are given a fractional vector $y \in [0, 1]^n$, for which $y(G) = 1$ for each $G \in \mathcal{G}$.

Let us introduce some further notation for knapsack-partition constraint systems. For each $j \in [n]$, we let $\mathcal{G}(j)$ denote the unique block $G \in \mathcal{G}$ with $j \in G$. For $W \subseteq [n]$, we define $\mathcal{G}(W) \subseteq \mathcal{G}$ to be the set of blocks involved in W, i.e. $\mathcal{G}(W) = \{\mathcal{G}(w) \mid w \in W\}$. For any set $\mathcal{G}_0 \subseteq \mathcal{G}$ and $W \subseteq [n]$, we define the set $W \wedge \mathcal{G}_0$ as

$$W \wedge \mathcal{G}_0 := W \cap \bigcup_{G \in \mathcal{G}_0} G = \{w \mid w \in W \text{ and } \mathcal{G}(w) \in \mathcal{G}_0\}.$$

For $G \in \mathcal{G}$, we define the following functions to count the number of fractional entries:

$$T_G(z) := \max(0, |\{i \in G \mid z_i \in (0, 1)\}| - 1).$$

We also define $T(z) := \sum_{G \in \mathcal{G}} T_G(z)$.

In an ideal scenario, we would like to perform a randomized rounding to generate a vector $Y \in \{0, 1\}^n$ satisfying the following desiderata (which are, to a certain

extent, mutually incompatible and unattainable – indeed, our algorithm instead guarantees certain related conditions (E1)–(E6)):

(D1) $\mathbf{E}[Y_j] = y_j$ for every $j \in [n]$.

(D2) The random variables Y_j are negatively correlated.

(D3) $Y(G) = 1$ for $G \in \mathcal{G}$.

(D4) $MY = My$.

To further understand these constraints, consider *independent selection*: independently for each block G, select exactly one item j, so that each item j is selected with probability y_j. This is a valid probability distribution as $y(G) = 1$ and the entries of y are in the range $[0, 1]$.

We write $Y = \textsc{IndSelect}(\mathcal{G}, y)$ to indicate that the (fully integral) vector Y is obtained by applying independent selection with respect to $y, \mathcal{G}$. It satisfies desiderata (D1), (D2), (D3) perfectly. However, it only weakly satisfies (D4): specifically, the value of $M_k Y$ will be a sum of negatively-correlated random variables with mean $M_k y$, which can deviate significantly from $M_k Y$.

2.1 The KPR algorithm

Now that we have defined general knapsack-partition systems, as well as the simplest rounding algorithm for them (independent selection), we are ready to define our main dependent rounding algorithm. The algorithm $\text{KPR}(\mathcal{G}, M, y, t)$ takes as input a set of blocks $\mathcal{G}$, a set of linear constraints M, a fractional vector y, and integer parameter t, and returns a mostly rounded vector $\tilde{Y} = \text{KPR}(\mathcal{G}, M, y, t)$.

The algorithm first applies a preprocessing step consisting of straightforward dependent rounding within each block G; we omit this here for space. It then repeatedly applies a more-complicated rounding process which modifies multiple blocks simultaneously, which we summarize as follows:

Algorithm 1 $\text{KPR}(\mathcal{G}, M, y, t)$

1: **while** $T(y) > t$ **do**
2: Form a set $\mathcal{J} \subseteq \mathcal{G}$, wherein each $G \in \mathcal{G}$ goes into $\mathcal{J}$ independently with probability $p = 3m/T(y)$.
3: **if** $\sum_{G \in \mathcal{J}} T_G(y) \geq m + 1$ **then**
4: Choose $\delta \in \mathbb{R}^n$ such that
* $M\delta = 0$, $y + \delta \in [0, 1]^n$, and $y - \delta \in [0, 1]^n$
* There is at least one index i with $y_i \in (0, 1)$ such that $y_i + \delta_i \in \{0, 1\}$ or $y_i - \delta_i \in \{0, 1\}$.
* $\delta_j = 0$ if $\mathcal{G}(j) \notin \mathcal{J}$
* $\delta(G) = 0$ for all $G \in \mathcal{G}$.
5: With probability $1/2$, update $y \leftarrow y + \delta$; else, update $y \leftarrow y - \delta$
6: **return** y

We refer to the loop at lines (1)–(3) of KPR as the IntraBlockReduce phase, and we write $y' = \textsc{IntraBlockReduce}(y)$ for the vector obtained after these steps. We write $y' = \text{KPR-ITERATION}(y)$ to denote a single iteration of the loop at lines (4)–(8); thus, lines (4)–(8) are equivalent to the following:

while $T(y) > t$ **do** update $y \leftarrow \text{KPR-ITERATION}(y)$

The KPR algorithm *requires throughout* that $t > 12m$; this assumption will not be stated explicitly again. Note that, because of this condition, the probability p in line (5) is at most $3m/t \leq 1/4$. (We remark that it is likely impossible to obtain fewer than m fractional entries, while still respecting the knapsack and partition constraints.)

2.2 KPR algorithm: formal results

The KPR algorithm generates a nearly-integral vector $\tilde{Y}$ satisfying a weaker form of negative correlation. We formulate this in terms of a potential function. For any set $W \subseteq [n]$ and vector $y \in [0, 1]^n$, we define

$$Q(W, y) := \prod_{G \in \mathcal{G}} (1 - y(W \cap G)).$$

As we have discussed, the desiderata (D1)–(D4) cannot be exactly satisfied. The vector $\tilde{Y} = \text{KPR}(\mathcal{G}, M, y, t)$ will instead satisfy the weaker constraints:

(E1) For all $W \subseteq [n]$, $\mathbf{E}[Q(W, \tilde{Y})]$ is "not much more than" $Q(W, y)$;

(E2) Every $j \in [n]$ has $\mathbf{E}[\tilde{Y}_j] = y_j$;

(E3) $\tilde{Y}(G) = 1$ for $G \in \mathcal{G}$;

(E4) $M\tilde{Y} = My$;

(E5) At most $2t$ entries of $\tilde{Y}$ are fractional;

(E6) For each block $G \in \mathcal{G}$, at most $m+1$ entries of $\tilde{Y}$ are fractional.

Here (E1) is intentionally vague because the relationship between $\mathbf{E}[Q(W,\tilde{Y})]$ and $Q(W,y)$ can be quite complex. Before we explain our result, let us explain the role played by the potential function Q. Note that for $Y = \textsc{IndSelect}(\mathcal{G}, y)$ we have $Y(W) = 0$ iff $Q(W,Y) = 1$ and hence

$$\Pr(Y(W) = 0) = \mathbf{E}[Q(W,Y)] = Q(W,y).$$

Thus, $\mathbf{E}[Q(W,\tilde{Y})]$ is a smoothed measure of whether the KPR output $\tilde{Y}$ satisfies $\tilde{Y}(W) = 0$.

Why might one be interested in upper-bounding terms of the form $\Pr(Y(W) = 0)$? We have briefly touched upon this earlier, but let us spell out in greater detail how such bounds arise naturally in clustering algorithms, such as our algorithms for knapsack median and knapsack center. These algorithms first cluster the facilities in some (greedy) manner (these are the blocks of $G \in \mathcal{G}$ in our notation), and open a facilities suitably at random from each cluster. Any given client j will first check if some "nearby" facility gets opened; if not, then it must use a "backup" facility which, however, is farther away. The bad event of no opened "nearby" facility is precisely modeled by events of the form $Y(W) = 0$. In some cases, the clustering algorithm may open multiple facilities; this can also be represented in terms of knapsack-partition systems, wherein the blocks $\mathcal{G}$ may correspond to all subsets of facilities to be opened in $\mathcal{G}$.

Our main result here covers a setting needed for a number of algorithmic applications: there is a small set of blocks $\mathcal{D} \subseteq \mathcal{G}$, such that $y(G \cap W)$ is close to one for $G \in \mathcal{D}$. We show the following:

Theorem 2.1 *Let $\mathcal{D} \subseteq \mathcal{G}$ with $|\mathcal{D}| = d$. Let $\tilde{Y} = \mathrm{KPR}(\mathcal{G}, M, y, t)$ with $t \geq 5000m(d+1)$. Then*

$$\mathbf{E}[Q(W,\tilde{Y})] \leq Q(W,y) + Q(W \wedge \mathcal{D}, y)(e^{O((d+1)^2 m^2/t)} - 1).$$

The result in Theorem 2.1 is complex and hard to use directly. We use it to derive a number of simplified results, such as the following two estimates:

Theorem 2.2 *The output $\tilde{Y}$ of KPR satisfies $\mathbf{E}[Q(W,\tilde{Y})] \leq Q(W,y) + O(m^2/t)$.*

Theorem 2.3 *Let $d = |\mathcal{G}(W)|$. For $t > 10000md$, the output $\tilde{Y}$ of KPR satisfies $\mathbf{E}[Q(W,\tilde{Y})] \leq Q(W,y)e^{O(m^2 d^2/t)}$.*

For simplicity, we do not attempt to optimize the constant factors here or elsewhere in the analysis.

3 A concentration inequality for additive knapsack pseudo-solutions

The second main technical tool is an intriguing connection between independent rounding and pseudo-additive solutions. For maximum generality, we state it in terms of a broader class of random variables satisfying a proper known as *negative association* [Joag-Dev and Proschan, 1983]. We provide the formal definition of this later; it includes independent random variables as a special case.

Theorem 3.1 *Let $X_1, \dots, X_n$ be negatively-associated, non-negative random variables. Then with probability at least $1 - \delta$, there is a set $W \subseteq [n]$ (which may depend on the values of $X_1, \dots, X_n$) with $|W| \leq O\left(\sqrt{n \log \frac{1}{\delta}}\right)$ such that*

$$\sum_{i \in [n] - W} X_i \leq \sum_{i \in [n]} \mathbf{E}[X_i].$$

We remark that Theorem 3.1 is tight for many values of (n, δ). For example, consider a system with n independent Bernoulli(p) variables for any constant $p \in (0,1)$. In this case, note that the set W will need to have cardinality $(\sum_i X_i) - np$, which will typically be on the order of $O\left(\sqrt{n \log \frac{1}{\delta}}\right)$ with probability δ.

Following [Joag-Dev and Proschan, 1983], the definition of negative association (NA) is given as follows:

Definition 3.2 (NA random variables) *Random variables $X_1, \dots, X_n$ are* negatively associated *(NA) if for every subset $A \subseteq [n]$, and any pair of non-decreasing functions f_1, f_2, the random variables $f_1(X_i : i \in A)$ and $f_2(X_j : j \in [n] - A)$ have non-positive covariance. (Here, "$f_1(X_i : i \in A)$" means f_1 applied to the tuple $(X_i : i \in A)$, and similarly for "$f_2(X_j : j \in [n] - A)$".)*

If $X_1, \dots, X_n$ are independent random variables, then they are NA. The class of NA random variables includes other random processes; for example, the load-vector of the urns in balls-and-urns processes [Dubhashi and Ranjan, 1998].

Notably, the bound of Theorem 3.1 does not depend on the variance or sizes of the variables $X_1, \dots, X_n$, which is quite different from conventional concentration bounds such as Chernoff-Hoeffding. This may be of independent interest beyond our rounding algorithm. By way of comparison, [Feige, 2006, Garnett, 2020, He et al., 2010, Samuels, 1966] give other concentration bounds for sums of nonnegative independent variables without regard to size or variance. Such results are useful for tasks such as probabilistic estimation

of network parameters and to hypergraph matchings: see, e.g., [Goldreich and Ron, 2008, Eden et al., 2017, Alon et al., 2012]. We anticipate that our tail bound will also have broader applications.

Let us briefly summarize the role of this concentration inequality in our overall rounding algorithm. As we have discussed, our dependent rounding algorithm stops with a vector $\tilde{Y} \in [0,1]^n$ which has a small number t of remaining fractional entries, and which satisfies the knapsack constraints exactly. At this point, we may apply a problem-specific "end-game" to convert $\tilde{Y}$ to a fully-integral solution Y.

One attractive option is to apply independent selection to the vector $\tilde{Y}$, ignoring the knapsack constraints. This may violate the knapsack constraints, but by how much? This is precisely the random process governed by our concentration inequality. The integral entries of $\tilde{Y}$ have no effect on this independent selection, so $\tilde{Y}$ effectively has t variables and and thus Y is an additive pseudo-approximation of order $\sqrt{t \log m}$.

We remark that our result is algorithmically "local" in that for some (efficiently computable) α, our approach is simply to make each i go into the small-cardinality "discarded set" $\mathcal{S}'$ iff $X_i > \alpha$.

4 Variants of KPR

The knapsack-partition setting and KPR algorithm are very general. We summarize here some simpler ways to use them, which will occur in a number of algorithmic scenarios. One natural rounding strategy for a knapsack-partition problem is to execute KPR up to some stage $t > 12m$, and then finish by independent rounding. Since this process comes up so frequently, let us define it formally as an algorithm FullKPR.

Algorithm 2 FullKPR$(\mathcal{G}, M, y, t)$

1: $\tilde{Y} \leftarrow \text{KPR}(\mathcal{G}, M, y, t)$
2: $Y \leftarrow \text{IndSelect}(\mathcal{G}, \tilde{Y})$
3: **return** Y

This resulting vector $Y \in \{0,1\}^n$ is *fully* integral; it will not exactly satisfy the knapsack constraints, but it will be relatively close (depending on the value of t). By combining our analysis of KPR, with our concentration inequality to handle the finishing IndSelect stage, we get the following:

Theorem 4.1 *Let Y = FullKPR$(\mathcal{G}, M, y, t)$ with $t > 12m$ and $y \in [0,1]^n$. Then, with probability at least $1-\delta$, the vector Y is a q-additive pseudo-solution to M with $q = O(\sqrt{t \log \frac{m}{\delta}})$. This probability bound holds even after conditioning on the fixed vector $\tilde{Y}$.*

Note that independent selection does not change the expectation of $Q(W, y)$; thus, all of our analysis of Q for KPR will carry over immediately to FullKPR. Thus, FullKPR also has good independence properties:

Proposition 4.2 *Let Y = FullKPR$(\mathcal{G}, M, y, t)$. For any set $W \subseteq [n]$ with $|W| = k$ and $t > 10000mk$, we have $\mathbf{E}[\prod_{j \in W} Y_j] \leq e^{O(m^2k^2/t)} \prod_{j \in W} y_j$.*

As a second example, dependent rounding can be interpreted as a special case of KPR. Given some vector $x \in [0,1]^v$, we form a knapsack-partition instance with $n = 2v$ by defining, for each item $i \in [v]$, the values $y_i = x_i, y_{i+v} = 1 - x_i$, and defining the block $G_i = \{i, v+i\}$. Also, given a system of m knapsack constraints M on $[v]$, we lift to this $[n]$ by setting $M_{i+v} = 0$ for $i \in [v]$.

We can then run KPR on this problem instance, and return the fractional vector $\tilde{X} \in [0,1]^v$ by setting $\tilde{X}_i = \tilde{Y}_i$ for $i \in [v]$. We let $\tilde{X} = \text{KPR-DepRound}(x, M, t)$ be the result of this process. Note that $\sum_{i=1}^{v} M_i \tilde{X}_i = \sum_{i=1}^{2v} M_i \tilde{Y}_i$. In this setting, we get a particularly crisp form for our near negative-correlation bounds:

Proposition 4.3 *Let S, T be disjoint subsets of $[n]$ and let $\tilde{X}$ = KPR-DepRound(x, M, t). For $t > 10000(|S| + |T|)m$, we have*

$$\mathbf{E}[\prod_{i \in S} \tilde{X}_i \prod_{i \in T} (1 - \tilde{X}_i)] \leq e^{O(\frac{m^2(|S|+|T|)^2}{t})} \prod_{i \in S} x_i \prod_{i \in T} (1 - x_i).$$

5 Discussion

We thus obtain improved approximation algorithms for clustering problems with "knapsack" and "partition" constraints, which model, for instance, key notions of fairness in clustering. Two main probabilistic techniques developed here are: (a) a new "Samuels-Feige" type of concentration inequality, which applies to *unbounded* random variables, and (b) a new dependent randomized-rounding approach, our KPR algorithm.

Acknowledgment. We thank the referees for their valuable comments. Thomas Pensyl and Khoa Trinh were supported in part by NSF award CCF-1422569, and by research awards from Adobe. Aravind Srinivasan was supported in part by NSF awards CCF-1422569 and CCF-1749864, and by research awards from Adobe, Amazon, and Google.

6 Acknowledgements

We thank Chandra Chekuri, Shi Li, Nikhil Bansal, and the referees of earlier versions for their helpful sugges-

tions. Research supported in part by NSF Awards CNS-1010789, CCF-1422569 and CCF-1749864, and by research awards from Adobe, Inc.

References

[Ageev and Sviridenko, 2004] Ageev, A. and Sviridenko, M. (2004). Pipage rounding: a new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328.

[Alon et al., 2012] Alon, N., Huang, H., and Sudakov, B. (2012). Nonnegative k-sums, fractional covers, and probability of small deviations. *Journal of Combinatorial Theory, Series B*, 102(3):784–796.

[Alon and Spencer, 2016] Alon, N. and Spencer, J. (2016). *The Probabilistic Method*. Wiley, 4th edition.

[Bansal, 2019] Bansal, N. (2019). On a generalization of iterated and randomized rounding. In *Proc. 51st ACM Symposium on Theory of Computing (STOC)*, pages 1125–1135.

[Bansal and Nagarajan, 2016] Bansal, N. and Nagarajan, V. (2016). Approximation-friendly discrepancy rounding. In *Proc. 18th international conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 375–386.

[Beck and Fiala, 1981] Beck, J. and Fiala, T. (1981). "Integer-making" theorems. *Discrete Applied Mathematics*, 3:1–8.

[Byrka et al., 2015] Byrka, J., Pensyl, T., Rybicki, B., Srinivasan, A., and Trinh, K. (2015). An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proc. 26th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 737–756.

[Byrka et al., 2010] Byrka, J., Srinivasan, A., and Swamy, C. (2010). Fault-tolerant facility location: a randomized dependent LP-rounding algorithm. In *Proc. 14th international conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 244–257.

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

[Charikar and Li, 2012] Charikar, M. and Li, S. (2012). A dependent LP-rounding approach for the k-median problem. *Automata, Languages, and Programming (ICALP)*, pages 194–205.

[Chekuri et al., 2010] Chekuri, C., Vondrák, J., and Zenklusen, R. (2010). Dependent randomized rounding via exchange properties of combinatorial structures. In *Proc. 51st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 575–584.

[Chekuri et al., 2011] Chekuri, C., Vondrák, J., and Zenklusen, R. (2011). Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proc. 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1080–1097.

[Chen et al., 2016] Chen, D. Z., Li, J., Liang, H., and Wang, H. (2016). Matroid and knapsack center problems. *Algorithmica*, 75(1):27–52.

[Dubhashi and Ranjan, 1998] Dubhashi, D. P. and Ranjan, D. (1998). Balls and bins: A study in negative dependence. *Random Structures & Algorithms*, 13(2):99–124.

[Eden et al., 2017] Eden, T., Ron, D., and Seshadhri, C. (2017). Sublinear time estimation of degree distribution moments: The degeneracy connection. In *Proc. 44th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 80, Article #7.

[Feige, 2006] Feige, U. (2006). On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM Journal on Computing*, 35(4):964–984.

[Gandhi et al., 2006] Gandhi, R., Khuller, S., Parthasarathy, S., and Srinivasan, A. (2006). Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53:324–360.

[Garnett, 2020] Garnett, B. (2020). Small deviations of sums of independent random variables. *Journal of Combinatorial Theory, Series A*, 169, Article #105119.

[Goldreich and Ron, 2008] Goldreich, O. and Ron, D. (2008). Approximating average parameters of graphs. *Random Structures & Algorithms*, 32(4):473–493.

[Harris et al., 2018] Harris, D. G., Li, S., Pensyl, T., Srinivasan, A., and Trinh, K. (2018). Approximation algorithms for stochastic clustering. In *Advances in Neural Information Processing Systems (NeurIPS) 31*, pages 6038–6047.

[Harris and Srinivasan, 2019] Harris, D. G. and Srinivasan, A. (2019). The Moser–Tardos framework with partial resampling. *Journal of the ACM*, 66(5):36.

[He et al., 2010] He, S., Zhang, J., and Zhang, S. (2010). Bounding probability of small deviation: a fourth moment approach. *Mathematics of Operations Research*, 35(1):208–232.

[Hochbaum and Shmoys, 1986] Hochbaum, D. S. and Shmoys, D. B. (1986). A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550.

[Joag-Dev and Proschan, 1983] Joag-Dev, K. and Proschan, F. (1983). Negative association of random variables with applications. *The Annals of Statistics*, pages 286–295.

[Karp et al., 1987] Karp, R. M., Leighton, F. T., Rivest, R. L., Thompson, C. D., Vazirani, U. V., and Vazirani, V. V. (1987). Global wire routing in two-dimensional arrays. *Algorithmica*, 2:113–129.

[Khuller et al., 2000] Khuller, S., Pless, R., and Sussmann, Y. J. (2000). Fault tolerant k-center problems. *Theoretical Computer Science*, 242(1-2):237–245.

[Krishnaswamy et al., 2011] Krishnaswamy, R., Kumar, A., Nagarajan, V., Sabharwal, Y., and Saha, B. (2011). The matroid median problem. In *Proc. 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1117–1130.

[Krishnaswamy et al., 2018] Krishnaswamy, R., Li, S., and Sandeep, S. (2018). Constant approximation for k-median and k-means with outliers via iterative rounding. In *Proc. 50th ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 646–659.

[Leighton et al., 2001] Leighton, F. T., Lu, C., Rao, S., and Srinivasan, A. (2001). New algorithmic aspects of the local lemma with applications to routing and partitioning. *SIAM Journal on Computing*, 31(2):626–641.

[Li and Svensson, 2016] Li, S. and Svensson, O. (2016). Approximating k-median via pseudo-approximation. *Journal of the ACM*, 45(2):530–547.

[Samuels, 1966] Samuels, S. M. (1966). On a Chebyshev-type inequality for sums of independent random variables. *The Annals of Mathematical Statistics*, 37(1):248–259.

[Srinivasan, 2001] Srinivasan, A. (2001). Distributions on level-sets with applications to approximation algorithms. In *Proc. 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 588–597.

[Srinivasan, 2006] Srinivasan, A. (2006). An extension of the Lovász Local Lemma, and its applications to integer programming. *SIAM Journal on Computing*, 36(3):609–634.