New results on information-theoretic clustering

Ferdinando Cicalese
Department of Computer Science,
University of Verona, Verona, Italy
ferdinano.cicalese@univr.it

Eduardo Laber
Departamento de Informática,
PUC-RIO, Rio de Janeiro, Brazil
laber@inf.puc-rio.br

Lucas Murtinho
Departamento de Informática,
PUC-RIO, Rio de Janeiro, Brazil
lucas.murtinho@gmail.com

Supplementary material — full version manuscript

Abstract

We study the problem of optimizing the clustering of a set of vectors when the quality of the clustering is measured by the Entropy or the Gini impurity measure. Our results contribute to the state of the art both in terms of best known approximation guarantees and inapproximability bounds: (i) we give the first polynomial time algorithm for Entropy impurity based clustering with approximation guarantee independent of the number of vectors and (ii) we show that the problem of clustering based on entropy impurity does not admit a PTAS. This also implies an inapproximability result in information theoretic clustering for probability distributions closing a problem left open in [Chaudhury and McGregor, COLT08] and [Ackermann et al., ECCC11]. We also report experiments with a new clustering method that was designed on top of the theoretical tools leading to the above results. These experiments suggest a practical applicability for our method, in particular, when the number of clusters is large.

1 Introduction

Data clustering is a fundamental tool in machine learning that is commonly used to reduce the computational resources required to analyse large datasets. For comprehensive descriptions of different clustering methods and their applications refer to [14, 15]. In general, clustering is the problem of partitioning a set of items so that, in the output partition, similar items are grouped together and dissimilar items are separated. When the items are represented as vectors that correspond to frequency counts or probability distributions many clustering algorithms rely on so called impurity measures (e.g., entropy) that estimate the dissimilarity of a group of items (see, e.g., [13] and references therein) In a simple example of this setting a company may want to group users according to their taste for different genres of movies. Each user u is represented by a vector, where the value of the ith component counts the number of times u watched movies from genre i. To evaluate the dissimilarity of a group of users we calculate the impurity of the sum of their associated vectors and then we select the partition for which the sum of the dissimilarities of its groups is minimum. The design of clustering methods based on impurity measures is the central theme of this paper.

Problem Description. An impurity measure $I : \mathbf{v} \in \mathbb{R}^g \mapsto I(\mathbf{v}) \in \mathbb{R}^+$ is a function that assigns to a vector $\mathbf{v}$ a non-negative value $I(\mathbf{v})$ so that the more homogeneous $\mathbf{v}$, with respect to the values of its coordinates, the larger its impurity. Well-known examples of impurity measures are the Entropy impurity (aka Information Gain in the context of random forests) and the Gini impurity [7]:

$$I_{Ent}(\mathbf{v}) = \|\mathbf{v}\|_1 \sum_{i=1}^{g} \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{\|\mathbf{v}\|_1}{v_i},$$

$$I_{Gini}(\mathbf{v}) = \|\mathbf{v}\|_1 \sum_{i=1}^{g} \frac{v_i}{\|\mathbf{v}\|_1} \left(1 - \frac{v_i}{\|\mathbf{v}\|_1}\right)$$

We are given a collection of n many g-dimensional vectors V with non-negative values and we are also given an impurity measure I. The goal is to find a partition $\mathcal{P}$ of V into k disjoint groups of vectors $V_1, \ldots, V_k$ so as to minimize the sum of the impurities of the groups in $\mathcal{P}$, i.e.,

$$I(\mathcal{P}) = \sum_{m=1}^{k} I\bigg(\sum_{\mathbf{v} \in V_m} \mathbf{v}\bigg). \tag{1}$$

We refer to this problem as the PARTITION WITH MINIMUM WEIGHTED IMPURITY PROBLEM (PMWIP). While we present results for I_{Gini} our main focus is on the Entropy impurity I_{Ent}. We use PMWIP$_{Ent}$ (PMWIP$_{Gini}$) to refer to PMWIP with impurity measure I_{Ent} (I_{Gini}).

Motivations. Clustering based on impurity measures is used in a number of relevant application as: (i) partition the values of attributes during the branching phase in the construction of random forest/decision trees [7, 8, 11, 12, 19]. (ii) clustering of words based on their distribution over a text collection for improving classification tasks [6, 13] and (iii) quantization of memoryless channels/design of polar codes [28, 18, 17, 25, 24]. Although these papers present their clustering optimization criterion in terms of different information theoretic concepts, e.g., mutual information, information gain, KL-divergence, we note that all of them can be rephrased in terms of our objective function, the entropy impurity measure. These equivalences are discussed in [11].

Despite of its wide use in relevant applications and entropy being, arguably, the most important measure in Information Theory as well as relevant in Machine Learning, the current understanding of PMWIP from the perspective of algorithms/complexity is very limited as we detail further. This contrasts with what is known for clustering in metric spaces where the gap between the ratios achieved by the best known algorithms and the largest known inapproximability factors, assuming $P \neq NP$, are somehow tight (see [5] and references therein). Our study contributes to change this scenario.

Our Results. First we present a simple linear time algorithm that simultaneously guarantees (i) an $O(\log \sum_{\mathbf{v} \in V} \|\mathbf{v}\|_1)$ approximation for PMWIP$_{Ent}$; (ii) an $O(\log n + \log g)$ approximation for the case where all vectors in V have the same ℓ_1 norm and (iii) a 3-approximation for the PMWIP$_{Gini}$. The last is tight in the sense that one cannot obtain a PTAS for PMWIP$_{Gini}$, unless P=NP, due to its connection with the k-means problem [5, 20].

Then, we present a second algorithm that provides an $O(\log^2(\min\{k, g\}))$-approximation for PMWIP$_{Ent}$ in polynomial time. Our algorithm is the first approximation algorithm for clustering based on entropy minimization, among those that do not rely on assumptions over the input data, which achieves an approximation that does not depends on n. We also explore a relation between vertex covers and star decompositions in cubic graphs to prove that PMWIP$_{Ent}$ is APX-Hard even for the case where all vectors have the same ℓ_1-norm. This result solves a problem that remained open in previous work [9, 2].

In order to assess the potential of our theoretical tools/findings for practical purposes we developed a new clustering method, on top of them, and compared it with Divisive Clustering [13], an adaptation of k-means that uses Kullback-Leibler divergence (KL-divergence) instead of squared Euclidean distance. We observe in our experiments, over two datasets, that the new method obtains partitions with impurity close to that obtained by Divisive Clustering. The advantage of our method is that it is much faster, especially when the number of clusters is large, since it runs in $O(n \log n + ng)$ time while Divisive Clustering has $\Theta(ngk)$ complexity per iteration.

Techniques. In terms of algorithmic techniques, when $g > k$, the first step of both algorithms proposed here is to employ an extension of the approach introduced in [19] that allows to reduce the dimensionality of the vectors in V from g to k with a controllable additive loss in the approximation ratio. In [19], where the case $k = 2$ is studied, after the reduction step, an optimal clustering algorithm is used. However, for arbitrary k, the focus of our work, the same strategy cannot be applied since the problem is NP-Complete. Thus, it is crucial to devise novel procedures to handle the case where $g \leq k$.

The procedure employed by the first algorithm is quite simple: it assigns vectors to groups according to the dominant coordinate, that is, one with the largest value. The procedure of the second algorithm is significantly more involved, it relies on the combination of the following results: (i) the existence of an optimal algorithm for $g = 2$ [18];(ii) the existence of a mapping $\chi : \mathbb{R}^g \mapsto \mathbb{R}^2$ such that for a set of vectors B which is pure, i.e., a set of vectors with the same dominant component, $I_{Ent}(\sum_{\mathbf{v} \in B} \mathbf{v}) = O(\log g) I_{Ent}(\sum_{\mathbf{v} \in B} \chi(\mathbf{v}))$ and (iii) a structural theorem that states that there exists a partition whose impurity is at an $O(\log^2 g)$ factor from the optimal one and such that at most one of its groups is mixed, i.e., it is not pure. The search for a partition of this type with low impurity can be achieved in pseudo-polynomial time via Dynamic Programming. To obtain a polynomial time algorithm we then employ a filtering technique similar to that employed for obtaining a FPTAS for the subset sum problem.

Related Work. We first discuss theoretical work on the problem. Kurkoski and Yagi [18] showed that PMWIP$_{Ent}$ can be solved in polynomial time when $g = 2$. The correctness of this algorithm relies on a theorem, proved in [7], which is generalized for $g > 2$ and k groups in [11, 8, 12]. These theorems state that there exists an optimal solution that can be separated by hyperplanes in $\mathbb{R}^g$. These results imply the existence of an $O(n^g)$ optimal algorithm when $k = 2$. Recently, it was proved that PMWIP$_{Ent}$ is NP-Complete, even when $k = 2$, and constant approximation algorithms were given for a class of impurity measures that includes Entropy and Gini for $k = 2$ [19]. As noted before their approach cannot be directly employed to handle the case where k is arbitrary.

PMWIP$_{Ent}$ has recently attracted large interest in the information theory community in the context of efficient quantizer design, and also motivated by the construction of polar codes [28, 18, 17, 25, 24] In our terminology, the focus of this series of work is proving bounds on the increase of impurity when we reduce the number of clusters from n to k.

PMWIP$_{Ent}$ is a generalization of MTC$_{KL}$ [9], the problem of clustering a set of n probability distributions into k groups minimizing the total Kullback-Leibler (KL) divergence from the distributions to the centroids of their assigned groups. MTC$_{KL}$ corresponds to the particular case of PMWIP$_{Ent}$ where each vector in V has the same ℓ_1 norm. While the optimal solutions of

PMWIP_{Ent} and MTC_{KL} match, the problems differ in terms of approximation since the objective function for MTC_{KL} has an additional constant term $-\sum_{\mathbf{v}\in V} I_{Ent}(\mathbf{v})$ so that an α-approximation for MTC_{KL} problem implies an α-approximation for PMWIP_{Ent} while the converse is not necessarily true.

In [9] an $O(\log n)$ approximation for MTC_{KL} is given. Some $(1+\epsilon)$-approximation algorithms were proposed for a constrained version of MTC_{KL} where every element of every probability distribution lies in the interval $[\lambda, v]$ [3, 1, 4, 22]. The algorithm from [3, 4] runs in $O(n2^{O(mk/\epsilon \log(mk/\epsilon))})$ time, where m is a constant that depends on ϵ and λ. In [1] the running time is improved to $O(ngk + g2^{O(k/\epsilon \log(k/\epsilon))} \log^{k+2}(n))$ via the use of weak coresets. Recently, using strong coresets, $O(ngk + 2^{poly(gk/\epsilon)}$ time is obtained [22]. We shall note that these algorithms provide guarantees for μ-similar Bregman divergences, a class of metrics that includes domain constrained KL divergence. By using similar assumptions on the components of the input probability distributions, Jegelka et. al. [16] show that Lloyds k-means algorithm—which also has an exponential time worst case complexity [29]—obtains an $O(\log k)$ approximation for MTC_{KL}.

Among the algorithms mentioned for MTC_{KL}, the one that allows a more direct comparison with ours is the method proposed in [9] since it runs in polytime and does not rely on assumptions over the input data. As discussed before an α-approximation for the MTC_{KL} problem implies α-approximation for the special case of PMWIP_{Ent} with vectors of the same ℓ_1 norm, so the approximation measure used in [9] is more challenging. However, our results apply to a more general problem and nonetheless we are able to provide approximation guarantee depending on the minimum between the logarithm of the number of clusters and the dimension while the guarantee in [9] depends on the logarithm of the number of input vectors.

In terms of computational complexity, Chaudhuri and McGregor [9] proved that a variant of MTC_{KL} where the centroids must be chosen from the vectors in V is NP-Complete. Ackermann et. al. [2] proved that MTC_{KL} is NP-Hard. Our hardness result for PMWIP_{Ent} implies that clustering with KL-Divergence if APX-Hard, improving the previous results.

Experimental work on clustering using impurity measures have been performed by a number of authors [6, 12, 27, 13, 21, 22]. A variant of Loyds k-means that uses Kullback-Leibler divergence rather than squared Euclidean distance was proposed independently in [11, 13]. Experiments from [13] suggest that this method, denoted by them as Divisive Clustering, is superior to those proposed in [6, 27]. That is the reason why we decided to compare our method with this specific one.

2 Preliminaries

We start defining some notations employed throughout the paper. An instance of PMWIP is a triple (V, k, I), where V is a collection of non-null vectors in $\mathbb{R}^g$ with non-negative integer coordinates, k is an integer larger than 1 and I is a scaled impurity measure.

We assume that for each coordinate $i = 1, \ldots, g$ there exists at least one vector $\mathbf{v} \in V$ whose ith coordinate is non-zero, i.e., the vector $\sum_{\mathbf{v}\in V} \mathbf{v}$ has no zero coordinates—for otherwise we could consider an instance of PMWIP with the vectors lying in some dimension $g' < g$. For a set of vectors S, the impurity $I(S)$ of S is given by $I(\sum_{\mathbf{v}\in S} \mathbf{v})$. The impurity of a partition $\mathcal{P} = (V^{(1)}, \ldots, V^{(k)})$ of the set V is then $I(\mathcal{P}) = \sum_{i=1}^{k} I(V^{(i)})$. We use $\text{OPT}(V, I, k)$ to denote the minimum possible impurity for an k-partition of V and, whenever the context is clear, we simply talk about instance V (instead of (V, I, k)) and of the impurity of an optimal solution as $\text{OPT}(V)$

(instead of $\mathrm{OPT}(V,I,k)$). We say that a partition $(V^{(1)},\ldots,V^{(k)})$ is optimal for input (V,I,k) iff $\sum_{i=1}^{k} I(V^{(i)}) = \mathrm{OPT}(V,I,k)$.

For an algorithm $\mathcal{A}$ and an instance (V,I,k), we denote by $\mathcal{A}(V,I,k)$ and $I(\mathcal{A}(V,I,k))$ the partition output by $\mathcal{A}$ on instance (V,I,k) and its impurity, respectively. Whenever it is clear from the context, we omit to specify the instance and write $I(\mathcal{A})$ for $I(\mathcal{A}(V,I,k))$.

We use bold face font to denote vectors, e.g., $\mathbf{u},\mathbf{v},\ldots$. For a vector $\mathbf{u}$ we use u_i to denote its ith component. Given two vectors $\mathbf{u}=(u_1,\ldots,u_g)$ and $\mathbf{v}=(v_1,\ldots,v_g)$ we use $\mathbf{u}\cdot\mathbf{v}$ to denote their inner product and $\mathbf{u}\circ\mathbf{v}=(u_1v_1,\ldots,u_gv_g)$ to denote their component-wise (Hadamard) product. We use $\mathbf{0}$ and $\mathbf{1}$ to denote the vectors in $\mathbb{R}^g$ with all coordinates equal to 0 and 1, respectively. We use $[m]$ to denote the set of the first m positive integers. For $i=1,\ldots,g$ we denote by $\mathbf{e}_i$ the vector in $\mathbb{R}^g$ with the ith coordinate equal to 1 and all other coordinates equal to 0.

The following properties will be useful in our analysis.

Proposition 1. *Let $p\in[0,0.5)$. Then, $p\log(1/p)\geq(1-p)\log[1/(1-p)]$*

Proof. For $p=0$ and $p=0.5$ the result holds. Moreover, the derivative of $p\log(1/p)-(1-p)\log[1/(1-p)]$ is $[\ln(1/(p)(1-p))-2]/\ln(2)$ □

Proposition 2. *Let $A>0$. The function $f(x)=x\log(A/x)$ is increasing in the interval $(0,A/e]$ and decreasing in the interval $(A/e,A]$ so that its maximum value in the interval $[0,A]$ is $(A\log e)/e$.*

Proof. The result follows because $f'(x)=(\ln(A)-\ln x-1)/\ln 2$, the derivative of $f(x)$ is positive in the interval $(0,A/e)$ and negative in the interval $[A/e,A)$. □

2.1 Frequency weighted impurity measures with subsystem property

The impurity measures we will focus on, namely Gini and Entropy, are special cases of a larger class of impurity measures, which we denote by $\mathcal{C}$, that satisfy the following definition

$$I(\mathbf{u})=\|\mathbf{u}\|_1\sum_{i=1}^{dim(\mathbf{u})} f\left(\frac{u_i}{\|\mathbf{u}\|_1}\right), \tag{P0}$$

where $dim(\mathbf{u})$ is the dimension of vector $\mathbf{u}$ and $f:\mathbb{R}\to\mathbb{R}$ is a function satisfying the following conditions:

1. $f(0)=f(1)=0$ (P1)
2. f is strictly concave in the interval [0,1] (P2)
3. For all $0<p\leq q\leq 1$, it holds that $f(p)\leq\frac{p}{q}\cdot f(q)+q\cdot f\left(\frac{p}{q}\right)$ (P3)

Impurity measures satisfying the conditions (P0)-(P2) are called *frequency-weighted impurity measures based on concave functions* [12]. A fundamental properties of such impurities measures is that they are *superadditive* as shown in [12]. We record this property in the following lemma.

Lemma 1 (Lemma 1 in [12])**.** *If I satisfies (P0)-(P2) then for every vectors $\mathbf{u}_k$ and $\mathbf{u}_R$ in $\mathbb{R}^g_+$, we have $I(\mathbf{u}_k+\mathbf{u}_R)\geq I(\mathbf{u}_k)+I(\mathbf{u}_R)$.*

The Entropy and the Gini impurity measure satisfy the definition (P0) by means of the functions $f_{Entr}(x) = -x \log x$ and $f_{Gini}(x) = x(1-x)$. In fact, for a vector $\mathbf{u} \in \mathbb{R}^g$ the Entropy impurity $I_{Ent}(\mathbf{u})$ and the Gini impurity $I_{Gini}(\mathbf{u})$ are defined by

$$I_{Ent}(\mathbf{u}) = \|\mathbf{u}\|_1 \sum_{i=1}^{g} f_{Entr}\left(\frac{u_i}{\|\mathbf{u}\|_1}\right) \quad \text{and} \quad I_{Gini}(\mathbf{u}) = \|\mathbf{u}\|_1 \sum_{i=1}^{g} f_{Gini}\left(\frac{u_i}{\|\mathbf{u}\|_1}\right). \tag{2}$$

It is also easy to see that $I_{Ent}(\mathbf{u}) = \|\mathbf{u}\|_1 H\left(\frac{u_1}{\|\mathbf{u}\|_1}, \frac{u_2}{\|\mathbf{u}\|_1}, \dots, \frac{u_g}{\|\mathbf{u}\|_1}\right)$ where $H(\cdot)$ denotes the Shannon entropy function.

The following fact states that both the Gini and Entropy impurity measures belong to the class $\mathcal{C}$. For the sake of self-containment we have deferred a simple proof of this fact to the appendix.

Fact 1. *Both f_{Entr} and f_{Gini} satisfy properties (P1)-(P3), and, in particular, we have that f_{Entr} satisfies (P3) with equality. Therefore both the Gini impurity measure I_{Gini} and the Entropy impurity measure I_{Ent} belong to $\mathcal{C}$.*

We now show that the impurity measures of class $\mathcal{C}$ satisfy a special *subsystem property* which will be used in our analysis to relate the impurity of partitions for instances of dimension g with the impurity of partitions for instances of dimension k.

Lemma 2 (Subsystem Property)**.** *Let I be an impurity measure in $\mathcal{C}$. Then, for every $\mathbf{u} \in \mathbb{R}^g_+$ and pairwise orthogonal vectors $\mathbf{d}^{(1)}, \dots, \mathbf{d}^{(k)} \in \{0,1\}^g$, such that $\sum_{i=1}^k \mathbf{d}^{(i)} = \mathbf{1}$, we have*

$$I(\mathbf{u}) \leq I\Big((\mathbf{u}\cdot\mathbf{d}^{(1)}, \mathbf{u}\cdot\mathbf{d}^{(2)}, \dots, \mathbf{u}\cdot\mathbf{d}^{(k)})\Big) + \sum_{i=1}^{k} I(\mathbf{u} \circ \mathbf{d}^{(i)}). \tag{3}$$

Moreover, for $I = I_{Ent}$ we have that (3) holds with equality.

Proof. Let f be the concave function used by the frequency-weighted impurity measure I.

For $i = 1, \dots, k$, let $\mathbf{u}^{(i)} = \mathbf{u} \circ \mathbf{d}^{(i)}$. We have

$$\begin{aligned}
I(\mathbf{u}) &= \|\mathbf{u}\|_1 \sum_{j=1}^{g} f(\frac{u_j}{\|\mathbf{u}\|_1}) & (4)\\
&= \|\mathbf{u}\|_1 \sum_{i=1}^{k} \sum_{j|d_j^{(i)}=1} f(\frac{u_j}{\|\mathbf{u}\|_1}) & (5)\\
&\leq \|\mathbf{u}\|_1 \sum_{i=1}^{k} \sum_{j|d_j^{(i)}=1} \frac{u_j}{\|\mathbf{u}\|_1} \frac{\|\mathbf{u}\|_1}{\|\mathbf{u}^{(i)}\|_1} f(\frac{\|\mathbf{u}^{(i)}\|_1}{\|\mathbf{u}\|_1}) + \frac{\|\mathbf{u}^{(i)}\|_1}{\|\mathbf{u}\|_1} f(\frac{u_j}{\|\mathbf{u}^{(i)}\|_1}) & (6)\\
&= \|\mathbf{u}\|_1 \sum_{i=1}^{k} \sum_{j|d_j^{(i)}=1} \frac{u_j}{\|\mathbf{u}^{(i)}\|_1} f(\frac{\|\mathbf{u}^{(i)}\|_1}{\|\mathbf{u}\|_1}) + \sum_{i=1}^{k} \sum_{j|d_j^{(i)}=1} \|\mathbf{u}^{(i)}\|_1 f(\frac{u_j}{\|\mathbf{u}^{(i)}\|_1}) & (7)\\
&= \|\mathbf{u}\|_1 \sum_{i=1}^{k} f(\frac{\|\mathbf{u}^{(i)}\|_1}{\|\mathbf{u}\|_1}) + \sum_{i=1}^{k} \|\mathbf{u}^{(i)}\|_1 \sum_{j|d_j^{(i)}=1} f(\frac{u_j}{\|\mathbf{u}^{(i)}\|_1}) & (8)\\
&= I\left((\mathbf{u}\cdot\mathbf{d}^{(1)}, \mathbf{u}\cdot\mathbf{d}^{(2)}, \dots, \mathbf{u}\cdot\mathbf{d}^{(k)})\right) + \sum_{i=1}^{k} I(\mathbf{u} \circ \mathbf{d}^{(i)}) & (9)
\end{aligned}$$

where (5) follows from (4) by splitting the second summation according to the partition of $[g]$ induced by the non zero components of the vectors $\mathbf{d}^{(i)}$; (6) follows from (5) by applying property (P3) with $p = \frac{u_j}{\|\mathbf{u}\|_1}$ and $q = \frac{\|\mathbf{u}^{(i)}\|_1}{\|\mathbf{u}\|_1}$; (7) follows from (6) by simple algebraic manipulations; (8) follows from (7) since by definition of $\mathbf{u}(i)$ we have $\sum_{j|d_j^{(i)}=1} u_j = \|\mathbf{u}^{(i)}\|_1$; (8) follows from (7) since $\|\mathbf{u}^{(i)}\|_1 = \mathbf{u} \cdot \mathbf{d}^{(i)}$ and $I(\mathbf{u} \circ \mathbf{d}^{(i)}) = \sum_{j|d_j^{(i)}=1} \|\mathbf{u} \circ \mathbf{d}^{(i)}\|_1 f(\frac{u_j}{\|\mathbf{u} \circ \mathbf{d}^{(i)}\|_1})$ and $\|\mathbf{u} \circ \mathbf{d}^{(i)}\|_1 = \|\mathbf{u}^{(i)}\|_1$.

The second statement of the lemma follows immediately by the fact that the concave function f_{Entr} satisfies property (P3) with equality (see Fact 1). Hence, for I_{Ent} the inequality in (6) becomes an equality.

$\square$

Remark 1. *The Subsystem property in the previous lemma holds also under the stronger assumption that vectors* $\mathbf{d}$*'s are from* $[0,1]^g$ *and not necessarily orthogonal.*

3 Handling high-dimensional vectors

In this section we present an approach to address instances (V, I, k) with $I \in \mathcal{C}$ and $g > k$. It consists of two steps: finding a 'good' projection of $\mathbb{R}^g$ into $\mathbb{R}^k$ and then solving PMWIP for the projected instance with $g = k$. Thus, in the next sections we will be focusing on how to build this projection and how to solve instances with $g \leq k$. The material of this section is a generalization for arbitrary k of the results introduced in [19] for $k = 2$.

Let $\mathcal{D}$ be the family of all sequences D of k pairwise orthogonal directions in $\{0,1\}^g$, such that $\sum_{\mathbf{d} \in D} \mathbf{d} = \mathbf{1}$. For each $D = (\mathbf{d}^{(1)}, \ldots, \cdot\mathbf{d}^{(k)}) \in \mathcal{D}$ and any $\mathbf{v} \in \mathbb{R}^g$ we define the operation $proj_D : \mathbb{R}^g_+ \to \mathbb{R}^k_+$ by

$$proj_D(\mathbf{v}) = (\mathbf{v} \cdot \mathbf{d}^{(1)}, \ldots, \mathbf{v} \cdot \mathbf{d}^{(k)}).$$

We also naturally extend the operation to sets of vectors S, by defining $proj_D(S)$ as the multiset of vectors obtained by applying $proj_D$ to each vector of S.

Let $\mathcal{A}$ be an algorithm that on instance (V, I, k) chooses a sequence of vectors $D = \{\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(k)}\} \in \mathcal{D}$ and returns a partition $(V^{(1)}, \ldots, V^{(k)})$ such that $(proj_D(V^{(1)}), \ldots, proj_D(V^{(k)}))$ is a 'good' partition for the k-dimensional instance $(proj_D(V), I, k)$. In this section we quantify the relationship between the approximation attained by $(proj_D(V^{(1)}), \ldots, proj_D(V^{(k)}))$ for instance $(proj_D(V), I, k)$ and the corresponding approximation attained by $(V^{(1)}, \ldots, V^{(k)})$ for instance (V, I, k).

Let $\mathbf{u} = \sum_{\mathbf{v} \in V} \mathbf{v}$ and $\mathbf{u}^{(i)} = \sum_{\mathbf{v} \in V^{(i)}} \mathbf{v}$. From the subsystem property we have the following upper bound on the impurity of the partition returned by $\mathcal{A}$.

$$I(\mathcal{A}) = \sum_{i=1}^{k} I(\mathbf{u}^{(i)}) \leq \sum_{i=1}^{k} I\left((\mathbf{u}^{(i)} \cdot \mathbf{d}^{(1)}, \ldots, \mathbf{u}^{(i)} \cdot \mathbf{d}^{(k)})\right) + \sum_{i=1}^{k} \sum_{\mathbf{d} \in D} I(\mathbf{u}^{(i)} \circ \mathbf{d})$$

Thus, by the superadditivity of I we have

$$I(\mathcal{A}) \leq \sum_{i=1}^{k} I\left((\mathbf{u}^{(i)} \cdot \mathbf{d}^{(1)}, \ldots, \mathbf{u}^{(i)} \cdot \mathbf{d}^{(k)})\right) + \sum_{\mathbf{d} \in D} I(\mathbf{u} \circ \mathbf{d}). \tag{10}$$

We now show two lower bounds on $\mathrm{OPT}(V, I, k)$. For the sake of simplifying the notation we will use $\mathrm{OPT}(V)$ for $\mathrm{OPT}(V, I, k)$.

Lemma 3. *For any instance* (V, I, k) *of* PMWIP *and any* $D = \{\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(k)}\} \in \mathcal{D}$ *we have* $\mathrm{OPT}(V) \geq \mathrm{OPT}(proj_D(V))$.

Proof. Let $V^{(1)}, \ldots, V^{(k)}$ be an optimal partition for V, i.e.,

$$\sum_{i=1}^{k} I(\sum_{\mathbf{v} \in V^{(i)}} \mathbf{v}) = \mathrm{OPT}(V). \tag{11}$$

We define the corresponding partition on the vectors $\tilde{\mathbf{v}}$ in $proj_D(V)$ by letting $\tilde{V}^{(i)} = \{proj_D(\mathbf{v}) \mid \mathbf{v} \in V^{(i)}\}$. We have

$$\sum_{i=1}^{k} I(\sum_{\tilde{\mathbf{v}} \in \tilde{V}^{(i)}} \tilde{\mathbf{v}}) \geq \mathrm{OPT}(proj_D(V)). \tag{12}$$

Let $\mathbf{u}^{(i)} = \sum_{\mathbf{v} \in V^{(i)}} \mathbf{v}$. Moreover, by the subadditivity of f, we have that for each $i = 1, \ldots, k$, it holds that

$$I(\sum_{\mathbf{v} \in V^{(i)}} \mathbf{v}) = ||\mathbf{u}^{(i)}||_1 \sum_{j=1}^{g} f\left(\frac{\mathbf{u}^{(i)}_j}{||\mathbf{u}^{(i)}||_1}\right) = ||\mathbf{u}^{(i)}||_1 \sum_{j=1}^{k} \sum_{\ell | d^{(j)}_\ell = 1} f\left(\frac{\mathbf{u}^{(i)}_\ell}{||\mathbf{u}^{(i)}||_1}\right) \geq$$

$$||\mathbf{u}^{(i)}||_1 \sum_{i=1}^{k} f\left(\frac{\sum_{\ell | d^{(j)}_\ell = 1} \mathbf{u}^{(i)}_\ell}{||\mathbf{u}^{(i)}||_1}\right) = I(\sum_{\tilde{\mathbf{v}} \in \tilde{V}^{(i)}} \tilde{\mathbf{v}})$$

which implies

$$\mathrm{OPT}(V) = \sum_{i=1}^{k} I(\sum_{\mathbf{v} \in V^{(i)}} \mathbf{v}) \geq \sum_{i=1}^{k} I(\sum_{\tilde{\mathbf{v}} \in \tilde{V}^{(i)}} \tilde{\mathbf{v}})$$

that combined with (12) gives the desired result. □

The following result, proved in [8, 12], states that the groups in the optimal solution can be separated by hyperplanes in $\mathbb{R}^k$. We recall it here as it will be used to derive our second lower bound on $\mathrm{OPT}(V)$ contained in Lemma 5 below.

Lemma 4 (Hyperplanes Lemma [8, 12])**.** *Let I be an impurity measure satisfying properties (P0)-(P2). If $(V_i)_{i=1,\ldots k}$ is an optimal partition of a set of vectors V, then there are vectors $\mathbf{v}^{(1)}, \ldots \mathbf{v}^{(k)} \in \mathbb{R}^g$ such that $\mathbf{v} \in V_i$ if and only if $\mathbf{v} \cdot \mathbf{v}^{(i)} < \mathbf{v} \cdot \mathbf{v}^{(j)}$ for each $j \neq i$.*

Lemma 5. *Let* (V, I, k) *be an instance of* PMWIP*. Let $\mathbf{u} = \sum_{\mathbf{v} \in V} \mathbf{v}$. It holds that*

$$\mathrm{OPT}(V) \geq \min_{D \in \mathcal{D}} \sum_{\mathbf{d}' \in D} I(\mathbf{u} \circ \mathbf{d}'),$$

Proof. Let W be the multiset of vectors built as follows: for each $\mathbf{v} = (v_1, \ldots, v_g) \in V$ we add the vectors $v_1\mathbf{e}_1, \ldots, v_g\mathbf{e}_g$ to W. Hence, W has ng vectors, all of them with only one non-zero coordinate.

It is not hard to see that for every partition $V^{(1)}, \ldots V^{(k)}$ of V there is a corresponding partition $W^{(1)}, \ldots, W^{(k)}$ such that $\sum_{i=1}^{k} I(\sum_{\mathbf{v} \in V^{(i)}} \mathbf{v}) = \sum_{i=1}^{k} I(\sum_{\mathbf{w} \in W^{(i)}} \mathbf{w})$, hence,

$$\mathrm{OPT}(V) \geq \mathrm{OPT}(W).$$

Let us now employ Lemma 4 to analyze $\mathrm{OPT}(W)$. Let $W^{(1)}, \ldots, W^{(k)}$, be a partition of W with impurity $\mathrm{OPT}(W)$. From Lemma 4 if two vectors $\mathbf{w}, \mathbf{w}' \in W$ are such that $\mathbf{w} = w\mathbf{e}_i$ and $\mathbf{w}' = w'\mathbf{e}_i$ for some i (i.e., they have the same non-zero component) then there is a j such that both $\mathbf{w}$ and $\mathbf{w}'$ belong to $W^{(j)}$.

For $j = 1, \ldots, k$, let $\mathbf{d}^{(j)}$ be the vector in $\{0,1\}^g$ such that $\mathbf{d}_i^{(j)} = 1$ if and only if the vectors of W whose only non-zero coordinate is the ith one are in $W^{(j)}$. Then $\{\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(k)}\} \in \mathcal{D}$ and we have

$$\mathrm{OPT}(W) = \sum_{i=1}^{k} I(\sum_{\mathbf{w} \in W} \mathbf{w} \circ \mathbf{d}^{(i)}) = \sum_{i=1}^{k} I(\sum_{\mathbf{v} \in V} \mathbf{v} \circ \mathbf{d}^{(i)}) = \sum_{i=1}^{k} I(\mathbf{u} \circ \mathbf{d}^{(i)}) \geq \min_{D \in \mathcal{D}} \sum_{\mathbf{d}' \in D} I(\mathbf{u} \circ \mathbf{d}').$$

□

Putting together (10) and Lemmas 3, 5 we have

$$\begin{aligned}
\frac{I(\mathcal{A})}{\mathrm{OPT}(V)} &\leq \frac{\sum_{i=1}^{k} I\left((\mathbf{u}^{(i)} \cdot \mathbf{d}^{(1)}, \ldots, \mathbf{u}^{(i)} \cdot \mathbf{d}^{(k)})\right) + \sum_{\mathbf{d} \in D} I(\mathbf{u} \circ \mathbf{d})}{\max\left\{\mathrm{OPT}(proj_D(V)), \min_{D \in \mathcal{D}} \sum_{\mathbf{d}' \in D} I(\mathbf{u} \circ \mathbf{d}')\right\}} \\
&\leq \frac{\sum_{i=1}^{k} I\left((\mathbf{u}^{(i)} \cdot \mathbf{d}^{(1)}, \ldots, \mathbf{u}^{(i)} \cdot \mathbf{d}^{(k)})\right)}{\mathrm{OPT}(proj_D(V))} + \frac{\sum_{\mathbf{d} \in D} I(\mathbf{u} \circ \mathbf{d})}{\min_{D \in \mathcal{D}} \sum_{\mathbf{d}' \in D} I(\mathbf{u} \circ \mathbf{d}')} \qquad (13)
\end{aligned}$$

Since the first ratio in the last expression is the approximation attained by the partition $proj_D(V^{(1)}), \ldots, proj_D(V^{(k)})$ on the instance $(proj_D(V), I, k)$, this inequality says that we can obtain a good approximation for instance (V, I, k) of PMWIP (where the vectors have dimension $g > k$) by properly choosing: (i) a set D of k orthogonal directions in $\{0,1\}^g$, and—given the choice of D—(ii) a good approximation for the instance $(proj_D(V), I, k)$, where the vectors have dimension k.

4 The dominance algorithm

For a vector $\mathbf{v}$ we say that i is the dominant component for $\mathbf{v}$ if $v_i \geq v_j$ for each $j \neq i$. In such a case we also say that $\mathbf{v}$ is i-dominant. For a set of vectors U we say that i is the dominant component in U if i is the dominant component for $\mathbf{u} = \sum_{\mathbf{v} \in U} \mathbf{v}$.

Given an instance (V, I, k) let $\mathbf{u} = \sum_{\mathbf{v} \in V} \mathbf{v}$ and let us assume that, up to reordering of the components, it holds that $u_i \geq u_{i+1}$, for $i = 1, \ldots, g-1$.

Let $\mathcal{A}^{Dom}$ be the algorithm that proceeds according to the following cases:

i $g > k$. $\mathcal{A}^{Dom}$ assigns each vector $\mathbf{v} = (v_1, \ldots, v_g) \in V$ to group i where i is the dominant component of vector $\mathbf{v}' = (v_1, \ldots, v_{k-1}, \sum_{j=k}^{g} v_j)$

ii $g \leq k$. $\mathcal{A}^{Dom}$ assigns each vector $\mathbf{v} \in V$ to group i where i is the dominant component of $\mathbf{v}$.

The only difference between cases (i) and (ii) is the reduction of dimensionality employed in the former to aggregate the smallest components with respect to $\mathbf{u}$.

Let $D = \{\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(k)}\} \in \mathcal{D}$ where $\mathbf{d}^{(i)} = \mathbf{e}_i$ for $i = 1, \ldots, k-1$ and $\mathbf{d}^{(k)} = \mathbf{1} - \sum_{\ell=1}^{k-1} \mathbf{d}^{(\ell)}$. We notice that that vector $\mathbf{v}'$ in case (i) is exactly $proj_D(\mathbf{v})$. Thus, if $g > k$, we can rewrite (13) as

$$\frac{I(\mathcal{A}^{Dom}(V))}{\mathrm{OPT}(V)} \leq \frac{I(\mathcal{A}^{Dom}(proj_D(V)))}{\mathrm{OPT}(proj_D(V))} + \frac{\sum_{\mathbf{d}\in D} I(\mathbf{u}\circ\mathbf{d})}{\min_{D\in\mathcal{D}}\sum_{\mathbf{d}'\in D} I(\mathbf{u}\circ\mathbf{d}')} \tag{14}$$

The next lemma is useful to prove an upper bound on the approximation of $\mathcal{A}^{Dom}$, when $g \leq k$.

Lemma 6. *Let (V, I, k) be an instance of* PMWIP *with $I \in \mathcal{C}$ and s.t. the dimension g of the vectors in V satisfies $g \leq k$. For a subset S of V let $\mathbf{u}^S = \sum_{\mathbf{v}\in S}\mathbf{v}$. If there exist positive numbers α, β such that for each $S \subseteq V$ we have*

$$\beta(\|\mathbf{u}^S\|_1 - \|\mathbf{u}^S\|_\infty) \leq I(\mathbf{u}^S) \leq \alpha(\|\mathbf{u}^S\|_1 - \|\mathbf{u}^S\|_\infty)$$

then the algorithm $\mathcal{A}^{Dom}$ guarantees α/β approximation, i.e.,

$$\frac{I(\mathcal{A}^{Dom}(V))}{\mathrm{OPT}(V)} \leq \frac{\alpha}{\beta}.$$

Proof. Let $(V^{(1)}, \ldots, V^{(k)})$ be the partition of V returned by $\mathcal{A}^{Dom}$. Then, by the superadditivity of I

$$\frac{I(\mathcal{A}^{Dom})}{\mathrm{OPT}(V)} = \frac{\sum_{i=1}^{g} I(\sum_{\mathbf{v}\in V^{(i)}}\mathbf{v})}{\sum_{i=1}^{g}\sum_{\mathbf{v}\in V^{(i)}} I(\mathbf{v})}.$$

Thus, it is enough to prove that for $i = 1, \ldots, g$

$$\frac{I(\sum_{\mathbf{v}\in V^{(i)}}\mathbf{v})}{\sum_{\mathbf{v}\in V^{(i)}} I(\mathbf{v})} \leq \frac{\alpha}{\beta}$$

Fix $i \in [g]$ and let $\mathbf{u} = \sum_{\mathbf{v}\in V^{(i)}}\mathbf{v}$. By hypothesis, we have

$$I(\mathbf{u}) \leq \alpha(\|\mathbf{u}\|_1 - \|\mathbf{u}\|_\infty) \quad \text{and} \quad I(\mathbf{v}) \geq \beta(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \text{ for every } \mathbf{v} \in V^{(i)}.$$

Moreover, by construction, for every vector $\mathbf{v} \in V^{(i)}$ we have $\|\mathbf{v}\|_\infty = v_i$, so that $\sum_{\mathbf{v}\in V^{(i)}}\|\mathbf{v}\|_\infty = \|\mathbf{u}\|_\infty$. Putting everything together we have

$$\frac{I(\sum_{\mathbf{v}\in V^{(i)}}\mathbf{v})}{\sum_{\mathbf{v}\in V^{(i)}} I(\mathbf{v})} \leq \frac{\alpha(\|\mathbf{u}\|_1 - \|\mathbf{u}\|_\infty)}{\sum_{\mathbf{v}\in V^{(i)}}\beta(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)} = \frac{\alpha(\|\mathbf{u}\|_1 - \|\mathbf{u}\|_\infty)}{\beta(\|\mathbf{u}\|_1 - \|\mathbf{u}\|_\infty)} = \frac{\alpha}{\beta},$$

as desired. □

4.1 Analysis of $\mathcal{A}^{Dom}$ for the Gini impurity measure I_{Gini}

In this section we show that algorithm $\mathcal{A}^{Dom}$ achieves constant 3-approximation when the impurity measure is I_{Gini}.

The following lemma together with Lemma 6 will show that $\mathcal{A}^{Dom}$ guarantees 2-approximation on instances with $g \leq k$.

Lemma 7. *For a vector $\mathbf{v} \in \mathbb{R}_+^g$ we have $\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty \leq I_{Gini}(\mathbf{v}) \leq 2(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)$.*

Proof. First we prove the upper bound. We have that

$$I_{Gini}(\mathbf{v}) = \|\mathbf{v}\|_1 \sum_{i=1}^{g} \frac{v_i}{\|\mathbf{v}\|_1}\left(1 - \frac{v_i}{\|\mathbf{v}\|_1}\right) = \frac{\sum_{i=1}^{g}\sum_{j\neq i} v_i v_j}{\|\mathbf{v}\|_1} \tag{15}$$

$$= \frac{\|\mathbf{v}\|_\infty(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) + \sum_{i|v_i\neq\|\mathbf{v}\|_\infty} v_i \sum_{j\neq i} v_j}{\|\mathbf{v}\|_1} \tag{16}$$

$$= \frac{2\|\mathbf{v}\|_\infty(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) + \sum_{i|v_i\neq\|\mathbf{v}\|_\infty} v_i \sum_{j\neq\|\mathbf{v}\|_\infty, j\neq i} v_j}{\|\mathbf{v}\|_1} \tag{17}$$

$$\leq \frac{2\|\mathbf{v}\|_\infty(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) + (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)^2}{\|\mathbf{v}\|_1} \tag{18}$$

$$= \frac{(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)(\|2\mathbf{v}\|_\infty + \mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)}{\|\mathbf{v}\|_1} \tag{19}$$

$$= \frac{(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)(\|\mathbf{v}\|_1 + \|\mathbf{v}\|_\infty)}{\|\mathbf{v}\|_1} \leq 2(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \tag{20}$$

For the lower bound we observe that $\sum_{i=1}^{g} v_i^2 \leq \|\mathbf{v}\|_\infty \|\mathbf{v}\|_1$. Therefore, we have

$$I_{Gini}(\mathbf{v}) = \|\mathbf{v}\|_1 \sum_{i=1}^{g} \frac{v_i}{\|\mathbf{v}\|_1}\left(1 - \frac{v_i}{\|\mathbf{v}\|_1}\right) = \|\mathbf{v}\|_1 - \frac{\sum_{i=1}^{g} v_i^2}{\|\mathbf{v}\|_1} \geq \|\mathbf{v}\|_1 - \frac{\|\mathbf{v}\|_\infty\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1} = \|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty. \tag{21}$$

□

Theorem 1. *Algorithm $\mathcal{A}^{Dom}$ is a 2-approximation algorithm for instances (V, I, k) with $I = I_{Gini}$ and $g \leq k$.*

Proof. Directly from Lemmas 6 and 7. □

The following lemma will provide an upper bound (in fact an exact estimate) of the second ratio in (14).

Lemma 8. *Fix a vector $\mathbf{u} \in \mathbb{R}^g$ such that $u_i \geq u_{i+1}$ for each $i = 1, \dots g-1$ and $D = \{\mathbf{d}^{(1)}, \dots \mathbf{d}^{(k)}\} \in \mathcal{D}$ with $\mathbf{d}^{(i)} = \mathbf{e}_i$ for $i = 1, \dots, k-1$ and $\mathbf{d}^{(k)} = \sum_{j=k}^{g} \mathbf{e}_j = \mathbf{1} - \sum_{j=1}^{k-1} \mathbf{d}^{(j)}$. It holds that*

$$\sum_{\mathbf{d}\in D} I(\mathbf{u}\circ\mathbf{d}) = \min_{D'\in\mathcal{D}} \left\{ \sum_{\mathbf{d}'\in D'} I(\mathbf{u}\circ\mathbf{d}') \right\}$$

Proof. Let $D^* \in \mathcal{D}$ be such that

$$\sum_{\mathbf{d}^*\in D^*} I(\mathbf{u}\circ\mathbf{d}^*) = \min_{D'\in\mathcal{D}} \left\{ \sum_{\mathbf{d}'\in D'} I(\mathbf{u}\circ\mathbf{d}') \right\} \tag{22}$$

and $|D^* \cap D|$ is maximum among all D^* satisfying (22).

Let us assume for the sake of contradiction that $D^* \neq D$. Let $\hat{\mathbf{d}} \in D^*$ such that $\hat{\mathbf{d}}_g = 1$. We note that $\hat{\mathbf{d}} \neq \mathbf{d}^{(k)}$ for otherwise we would have $D^* = D$.

Let $\mathbf{c} \in D^* \setminus (D \cup \{\hat{\mathbf{d}}\})$ such that for all other $\mathbf{d} \in D^* \setminus (D \cup \{\hat{\mathbf{d}}\})$ we have $\min\{i \mid \mathbf{c}_i = 1\} < \min\{i \mid \mathbf{d}_i = 1\}$, i.e., $\mathbf{c}$ is the vector in $D^* \setminus (D \cup \{\hat{\mathbf{d}}\})$ with the smallest non-zero component.

Let $\mathbf{v} = \mathbf{c} + \hat{\mathbf{d}}$ and i^* be the minimum integer such that $\mathbf{v}_{i^*} = 1$. Note that $i^* \leq k-1$, for otherwise we would have $D^* \notin \mathcal{D}$. Let F be the set of vectors from $\mathcal{D}$ defined by

$$F = (D^* \setminus \{\hat{\mathbf{d}}, \mathbf{c}\}) \cup \{\mathbf{d}^{(i^*)}, \mathbf{v} - \mathbf{d}^{(i^*)}\}.$$

The following claim directly follows from [19, Lemma 4.1]. For the sake of self-containment we defer its proof to the appendix.

Claim. Fix $\mathbf{u} \in \mathbb{R}^g$ such that $u_i \geq u_{i+1}$ for each $i = 1, \ldots, g-1$. Let $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$ two orthogonal vectors from $\{0,1\}^g \setminus \{\mathbf{0}\}$. Let $i^* = \min\{i \mid \max\{z_i^{(1)}, z_i^{(2)}\} = 1\}$ and $\mathbf{v}^{(1)} = \mathbf{e}_{i^*}$ and $\mathbf{v}^{(2)} = \mathbf{z}^{(1)} + \mathbf{z}^{(2)} - \mathbf{e}_{i^*}$. Then

$$I(\mathbf{u} \circ \mathbf{v}^{(1)}) + I(\mathbf{u} \circ \mathbf{v}^{(2)}) \leq I(\mathbf{u} \circ \mathbf{z}^{(1)}) + I(\mathbf{u} \circ \mathbf{z}^{(1)}).$$

By the Claim, we have that

$$\begin{aligned}
\sum_{\mathbf{d} \in F} I(\mathbf{u} \circ \mathbf{d}) &= I(\mathbf{u} \circ \mathbf{d}^{(i^*)}) + I(\mathbf{u} \circ (\mathbf{v} - \mathbf{d}^{(i^*)})) + \sum_{\mathbf{d} \in F \cap D^*} I(\mathbf{u} \circ \mathbf{d}) \\
&\leq I(\mathbf{u} \circ \hat{\mathbf{d}}) + I(\mathbf{u} \circ \mathbf{c}) + \sum_{\mathbf{d} \in F \cap D^*} I(\mathbf{u} \circ \mathbf{d}) = \sum_{\mathbf{d} \in D^*} I(\mathbf{u} \circ \mathbf{d}),
\end{aligned}$$

hence since D satisfies (22) we have that F also satisfies (22).

In addition we observe that $|D \cap F| > |D \cap D^*|$ as by definition it shares with D all that was shared by D^* and also $\mathbf{d}^{(i^*)}$. This would be in contradiction with the maximality of the intersection of D^*. Therefore, we must have $D^* = D$ which concludes the proof. □

Putting together inequalities (14), Theorem 1 and Lemma 8 we get that

Theorem 2. *Algorithm $\mathcal{A}^{Dom}$ is a linear time 3-approximation for the I_{Gini}*

4.2 Analysis of $\mathcal{A}^{Dom}$ for the Entropy impurity measure I_{Ent}

The following lemma will be useful for applying Lemma 6 to the analysis of the performance of $\mathcal{A}^{Dom}$ with respect to the entropy impurity measure I_{Ent}.

Lemma 9. *For a vector $\mathbf{v} \in \mathbb{R}^g_+$ we have*

$$(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log\left(\frac{\|\mathbf{v}\|_1}{\min\{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty, \|\mathbf{v}\|_\infty\}}\right) \leq I_{Ent}(\mathbf{v}) \leq 2(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log\left(\frac{g\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty}\right)$$

.

Proof. Let i^* be an index in $[g]$ such that $v_{i^*} = \|\mathbf{v}\|_\infty$. We have that

$$\begin{aligned}
I_{Ent}(\mathbf{v}) &= \|\mathbf{v}\|_\infty \log \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_\infty} + \sum_{i \neq i^*} v_i \log \frac{\|\mathbf{v}\|_1}{v_i} && (23) \\
&= \|\mathbf{v}\|_\infty \log \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_\infty} + (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \|\mathbf{v}\|_1 - \sum_{i \neq i*} v_i \log v_i. && (24)
\end{aligned}$$

For the upper bound, we observe that the expression in (24) is maximum when $v_i = (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)/(g-1)$ for $i \neq i^*$. Thus,

$$I_{Ent}(\mathbf{v}) \leq \|\mathbf{v}\|_\infty \log \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_\infty} + (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{\|\mathbf{v}\|_1}{(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)} + (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log(g-1). \quad (25)$$

To show that this satisfies the desired upper bound, we split the analysis into two cases:
If $\|\mathbf{v}\|_\infty \geq \frac{\|\mathbf{v}\|_1}{2}$ we have that

$$\begin{aligned} I_{Ent}(\mathbf{v}) &\leq \|\mathbf{v}\|_\infty \log \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_\infty} + (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty} + (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log(g-1) \\ &\leq 2(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{\|\mathbf{v}\|_1}{(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)} + (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log(g-1) \\ &\leq 2(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{g\|\mathbf{v}\|_1}{(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)}, \end{aligned}$$

where the second inequality follows from Proposition 1 using $p = (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)/\|\mathbf{v}\|_1$.
If $\|\mathbf{v}\|_\infty \leq \frac{\|\mathbf{v}\|_1}{2}$ we have that

$$\begin{aligned} I_{Ent}(\mathbf{v}) &\leq 2(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \frac{\log e}{e} + (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty} + (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log(g-1) \\ &\leq 2(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{g\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty}. \end{aligned}$$

where the first inequality follows from (25) and Proposition 2.

For the lower bound, consider the same two cases:
If $\|\mathbf{v}\|_\infty > \frac{\|\mathbf{v}\|_1}{2}$, the expression in (24) is minimum when there is a unique index $j \neq i^*$ such that $v_j = \|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty$ and $v_i = 0$ for each $i \in [g] \setminus \{j, i^*\}$. Thus,

$$I_{Ent}(\mathbf{v}) \geq \|\mathbf{v}\|_\infty \log \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_\infty} + (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty} \geq (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{\|\mathbf{v}\|_1}{\min\{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty, \|\mathbf{v}\|_\infty\}}$$

If $\|\mathbf{v}\|_\infty < \frac{\|\mathbf{v}\|_1}{2}$, the expression in (24) is minimum when there exists a set of indexes $A \subseteq [g]$ with $|A| = \lceil \|\mathbf{v}\|_1 / \|\mathbf{v}\|_\infty \rceil - 1$ such that $v_i = \|\mathbf{v}\|_\infty$ for each $i \in A$ and (possibly) an index $j \notin A$ such that $v_j = \|\mathbf{v}\|_1 - |A| \cdot \|\mathbf{v}\|_\infty$. Thus,

$$I_{Ent}(\mathbf{v}) \geq (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_\infty} \geq (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{\|\mathbf{v}\|_1}{\min\{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty, \|\mathbf{v}\|_\infty\}}$$

□

From the bounds in the previous lemma and Lemma 6 we obtain our first guarantee on the approximation of algorithm $\mathcal{A}^{Dom}$ for the Entropy Impurity measure on instances with $g \leq k$.

Theorem 3. *Let (V, I, k) be an instance of* PMWIP *with $I = I_{Ent}$ and $g \leq k$. Let $p = \log g + \log(\sum_{\mathbf{v} \in V} \|\mathbf{v}\|_1))$. Then, $\mathcal{A}^{Dom}$ guarantees a $2p$-approximation on instance (V, I, k).*

Proof. Let S be a subset of V and let $\mathbf{u}^S = \sum_{\mathbf{v} \in S} \mathbf{v}$. Define $\alpha = 1$ and $\beta = 2p$.

If $\|\mathbf{u}^S\|_1 = \|\mathbf{u}^S\|_\infty$ then $I(\mathbf{u}^S) = 0$ so that the conditions of Lemma 6 is satisified. Otherwise, Lemma 9 guarantees that

$$\|\mathbf{u}^S\|_1 - \|\mathbf{u}^S\|_\infty \leq I(\mathbf{u}^S) \leq 2(\|\mathbf{u}^S\|_1 - \|\mathbf{u}^S\|_\infty) \log(g\|\mathbf{u}^S\|_1) \leq \|\mathbf{u}^S\|_1 - \|\mathbf{u}^S\|_\infty 2p.$$

Thus, it follows from Lemma 6 that we have a $2p-$approximation. □

Remark 2. *Let s be a large integer. The instance $\{(s,0),(2,1),(0,1)\}$ and $k=2$ shows that the analysis is tight up to constant factors. In fact, the impurity of $\mathcal{A}^{Dom}$ is larger than $\log s$ while the impurity of the partition that leaves $(s,0)$ alone is 4.*

Theorem 4. *Let Uniform-*PMWIP *(*U-PMWIP*) be the variant of* PMWIP *where all vectors have the same ℓ_1 norm. We have that $\mathcal{A}^{Dom}$ is an $O(\log n + \log g)$-approximation algorithm for* U-PMWIP *with $I = I_{Ent}$ and $g \leq k$.*

Proof. Let (V, I, k) be an instance of U-PMWIP with $I = I_{Ent}$ and vectors of dimension $g \leq k$. Let $(V^{(1)}, \ldots, V^{(k)})$ be the partition of V returned by $\mathcal{A}^{Dom}$. By the superadditivity of I it holds that

$$\frac{I(\mathcal{A}^{Dom})}{\mathrm{OPT}(V)} = \frac{\sum_{i=1}^{g} I(\sum_{\mathbf{v} \in V^{(i)}} \mathbf{v})}{\sum_{i=1}^{g} \sum_{\mathbf{v} \in V^{(i)}} I(\mathbf{v})}$$

Thus, it is enough to prove that for $i = 1, \ldots, g$

$$\frac{I(\sum_{\mathbf{v} \in V^{(i)}} \mathbf{v})}{\sum_{\mathbf{v} \in V^{(i)}} I(\mathbf{v})} = O(\log n + \log g)$$

Let s be the ℓ_1 norm of all vectors in V, let $\mathbf{u} = \sum_{\mathbf{v} \in V^{(i)}} \mathbf{v}$ and let $c = \|\mathbf{u}\|_1 - \|\mathbf{u}\|_\infty$. By Lemma 9, we have that

$$I(\mathbf{u}) \leq c \log \frac{gns}{c}.$$

Moreover, we have

$$\sum_{\mathbf{v} \in V^{(i)}} I(\mathbf{v}) \geq \max \left\{ c, c \log s - \sum_{\mathbf{v} \in V^{(i)}} (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \right\}$$

If $c \geq s/2$ then we have a $O(\log n + \log g)$ approximation using c as a lower bound. If $c < s/2$ we get that

$$\sum_{\mathbf{v} \in V^{(i)}} I(\mathbf{v}) \geq c \log s - c \log c = c \log(s/c)$$

and the approximation is $O(\log n + \log g)$ as well □

Remark 3. *Let s be a large integer. The instance with $n-1$ vectors equal to $(s,0)$, one vector equals to $(s, s/2)$ and $k=2$ shows that the analysis is tight.*

To obtain an approximation of $\mathcal{A}^{Dom}$ for I_{Ent} for general k and g we need an upper bound on the second fraction in Equation (13). This is given by the next lemma.

Lemma 10. *Fix a vector* $\mathbf{u} \in \mathbb{R}^g$ *such that* $u_i \geq u_{i+1}$ *for each* $i = 1, \ldots g-1$ *and* $D = \{\mathbf{d}^{(1)}, \ldots \mathbf{d}^{(k)}\} \in \mathcal{D}$ *with* $\mathbf{d}^{(i)} = \mathbf{e}_i$ *for* $i = 1, \ldots, k-1$ *and* $\mathbf{d}^{(k)} = \sum_{j=k}^{g} \mathbf{e}_j = \mathbf{1} - \sum_{j=1}^{k-1} \mathbf{d}^{(j)}$. *It holds that*

$$\sum_{\mathbf{d} \in D} I_{Ent}(\mathbf{u} \circ \mathbf{d}) \leq O(\log k) \min_{D' \in \mathcal{D}} \left\{ \sum_{\mathbf{d}' \in D'} I_{Ent}(\mathbf{u} \circ \mathbf{d}') \right\}$$

Proof. Let $D^* = \{\mathbf{d}_*^{(1)}, \ldots, \mathbf{d}_*^{(k)}\} \in \mathcal{D}$ be such that

$$\sum_{\mathbf{d} \in D^*} I_{Ent}(\mathbf{u} \circ \mathbf{d}) = \min_{D' \in \mathcal{D}} \left\{ \sum_{\mathbf{d}' \in D'} I_{Ent}(\mathbf{u} \circ \mathbf{d}') \right\} \tag{26}$$

and $|D \cap D^*|$ is maximum among all set of vectors in $\mathcal{D}$ satisfying (26). Assume that $D \neq D^*$ for otherwise the claim holds trivially.

By Lemma 2 we have that for every $\hat{D} = \{\hat{\mathbf{d}}^{(1)}, \ldots, \hat{\mathbf{d}}^{(k)}\} \in \mathcal{D}$

$$\begin{aligned}
\sum_{i=1}^{k} I_{Ent}(\mathbf{u} \circ \hat{\mathbf{d}}^{(i)}) &= I_{Ent}(\mathbf{u}) - I_{Ent}(\mathbf{u} \cdot \hat{\mathbf{d}}^{(1)}, \ldots, \mathbf{u} \cdot \hat{\mathbf{d}}^{(k)}) \\
&= \|\mathbf{u}\|_1 \left(H\left(\frac{u_1}{\|\mathbf{u}\|_1}, \ldots, \frac{u_g}{\|\mathbf{u}\|_1} \right) - H\left(\frac{\mathbf{u} \cdot \hat{\mathbf{d}}^{(1)}}{\|\mathbf{u}\|_1}, \ldots, \frac{\mathbf{u} \cdot \hat{\mathbf{d}}^{(k)}}{\|\mathbf{u}\|_1} \right) \right)
\end{aligned}$$

where $H()$ denotes the Entropy function. Let us define $H(\hat{D}) = H\left(\frac{\mathbf{u} \cdot \hat{\mathbf{d}}^{(1)}}{\|\mathbf{u}\|_1}, \ldots, \frac{\mathbf{u} \cdot \hat{\mathbf{d}}^{(k)}}{\|\mathbf{u}\|_1} \right)$

Then $\hat{D}$ is a set of vectors that minimizes $\sum_{i=1}^{k} I_{Ent}(\mathbf{u} \circ \hat{\mathbf{d}}^{(i)})$ iff it maximizes $H(\hat{D})$.

We can think of the vectors in $\hat{D}$ as buckets containing components of $\mathbf{u}$, and we say that u_j is in bucket i if $\hat{d}_j^{(i)} = 1$. From the above formula and the concavity property of the Entropy function we have that the following claim holds.

Claim 1. Assume that there exists a subset $A \subseteq \{j \mid \hat{d}_j^{(i)} = 1\}$ of bucket i and a subset $B \subseteq \{j' \mid \hat{d}_{j'}^{(i')} = 1\}$ of bucket i' such that

$$\left| \left(\hat{\mathbf{d}}^{(i)} \cdot \mathbf{u} - \sum_{j \in A} u_j + \sum_{j' \in B} u_{j'} \right) - \left(\hat{\mathbf{d}}^{(i')} \cdot \mathbf{u} - \sum_{j' \in B} u_{j'} + \sum_{j \in A} u_j \right) \right| \leq |\hat{\mathbf{d}}^{(i)} \cdot \mathbf{u} - \hat{\mathbf{d}}^{(i')} \cdot \mathbf{u}| \tag{27}$$

i.e., swapping bucket for elements in A and B does not increase the absolute difference between the sum of elements in buckets i and i'. Then, for the set of vectors $\tilde{D} = \{\tilde{\mathbf{d}}^{(1)}, \ldots, \tilde{\mathbf{d}}^{(k)}\} \in \mathcal{D}$ defined by

$$\tilde{\mathbf{d}}^{(\ell)} = \begin{cases} \hat{\mathbf{d}}^{(\ell)} & \ell \notin \{i, i'\} \\ \hat{\mathbf{d}}^{(i)} - \sum_{j \in A} \mathbf{e}_j + \sum_{j' \in B} \mathbf{e}_{j'} & \ell = i \\ \hat{\mathbf{d}}^{(\ell)} - \sum_{j' \in B} \mathbf{e}_{j'} + \sum_{j \in A} \mathbf{e}_j & \ell = i', \end{cases}$$

i.e., for the set of vectors corresponding to the new buckets, it holds that $H\left(\frac{\mathbf{u} \cdot \hat{\mathbf{d}}^{(1)}}{\|\mathbf{u}\|_1}, \ldots, \frac{\mathbf{u} \cdot \hat{\mathbf{d}}^{(k)}}{\|\mathbf{u}\|_1} \right) \leq H\left(\frac{\mathbf{u} \cdot \tilde{\mathbf{d}}^{(1)}}{\|\mathbf{u}\|_1}, \ldots, \frac{\mathbf{u} \cdot \tilde{\mathbf{d}}^{(k)}}{\|\mathbf{u}\|_1} \right)$, with the equality holding iff inequality (27) is tight.

Because of Claim 1, we have that D^* satisfying (26) is a set of vectors that coincides with buckets that distribute the components of $\mathbf{u}$ in *the most balanced way*, i.e., $H(D^*)$ is maximum among all $D \in \mathcal{D}$.

From these observations, we can characterize the structure of buckets of D^*. For the sake of a simpler notation, let us denote with $S^{(i)}$ the sum of components in bucket $\mathbf{d}_*^{(i)}$, i.e., $S^{(i)} = \mathbf{u} \cdot \mathbf{d}_*^{(i)}$. We have the following

Claim 2. The set D^* satisfies the following properties:

(i) there is no bucket i that consists of a single element u_j with $j \geq k$;

(ii) if u_j is not alone in bucket i then for each $i' \neq i$ it holds that $S^{(i')} \geq u_j$;

(iii) if u_j is not alone in bucket i then for each $i' \neq i$ it holds that $S^{(i')} \geq S^{(i)} - u_j$;

For (i), assume, by contradiction that such i and j exists. Then, since $D^* \neq D$, there exists a bucket $i' \neq i$ that contains at least two elements, with one of them being $u_{j'}$ for some $j' < k$. Then, by Claim 1, swapping the buckets for u_j and $u_{j'}$ produces a new set of vectors with entropy not smaller than $H(D^*)$ and intersection with D larger than that of D^*, which is a contradiction.

For (ii), we observe that if there exists a bucket i' such that $S^{(i')} < u_j$ by moving every element but u_j from bucket i to bucket i', by Claim 1, we get a new set of vectors with entropy larger than $H(D^*)$, which is a contradiction.

For (iii), we observe that if there exists a bucket i' such that $S^{(i')} < S^{(i)} - u_j$ then by moving u_j from bucket i to bucket i', by Claim 1, we get a new set of vectors with entropy larger than $H(D^*)$, which is a contradiction.

We are now ready to prove the statement of the lemma. We have that $I_{Ent}(\mathbf{u} \circ \mathbf{d}^{(i)}) = 0$ for $i = 1, \ldots, k-1$. This holds since buckets $1, \ldots, k-1$ contain only one element. Let $S = \sum_{j \geq k} u_j$, and define $i(j)$ to be the bucket of D^* that contains u_j, for each $j = 1, \ldots, g$. We have

$$\sum_{i=1}^{k} I_{Ent}(\mathbf{u} \circ \mathbf{d}^{(i)}) = I_{Ent}(\mathbf{u} \circ \mathbf{d}^{(k)}) = \sum_{j \geq k} u_j \log \frac{S}{u_j} = \sum_{\substack{j \geq k \\ u_j \leq S^{(i(j))}/2}} u_j \log \frac{S}{u_j} + \sum_{\substack{j \geq k \\ u_j > S^{(i(j))}/2}} u_j \log \frac{S}{u_j} \quad (28)$$

where in the last expression we split the summands according to whether $u_j \geq S^{(i(j))}/2$ or $u_j < S^{(i(j))}/2$. We will argue that

$$\sum_{\substack{j \geq k \\ u_j \leq S^{(i(j))}/2}} u_j \log \frac{S}{u_j} \quad \text{is} \quad O(\log k) \sum_{i=1}^{k} I_{Ent}(\mathbf{u} \circ \mathbf{d}_*^{(i)}) \quad (29)$$

$$\sum_{\substack{j \geq k \\ u_j > S^{(i(j))}/2}} u_j \log \frac{S}{u_j} \quad \text{is} \quad O(\log k) \sum_{i=1}^{k} I_{Ent}(\mathbf{u} \circ \mathbf{d}_*^{(i)}), \quad (30)$$

from which the statement of the lemma follows.

Proof of Inequality (29).

Since

$$\sum_{i=1}^{k} I_{Ent}(\mathbf{u} \circ \mathbf{d}_*^{(i)}) = \sum_{j=1}^{g} u_j \frac{S^{(i(j))}}{u_j} \geq \sum_{\substack{j \geq k \\ u_j \leq S^{(i(j))}/2}} u_j \log \frac{S^{(i(j))}}{u_j}. \tag{31}$$

it is enough to show that for each $j \geq k$, with $u_j \leq S^{(i(j))/2}$, we have

$$\frac{u_j \log(S/u_j)}{u_j \log(S^{(i(j))}/u_j)} \leq \log(4k). \tag{32}$$

The above inequality can be established by showing that $S \leq 2k \cdot S^{(i(j))}$ and, then, using the bound $\frac{\log a}{\log b} \leq \log(2a/b)$, which holds whenever $b \geq 2$ and $a \geq b$.

To see that $S \leq 2k \cdot S^{(i(j))}$, let ℓ be a bucket in D^* containing some $u_{j'}$ for $j' \geq k$. By Claim 2, item (i), we have that bucket ℓ contains at least two elements. Let $e(\ell)$ be the element in bucket ℓ of minimum value. Then, by Claim 2, item (iii), we have

$$S^{(i(j))} \geq S^{(\ell)} - u_{e(\ell)} \geq S^{(\ell)}/2, \tag{33}$$

where the last inequality follows from the fact that bucket ℓ has at least two elements. Let $B = \{\ell \mid$ bucket ℓ has at least one element $u_{j'}$ with $j' \geq k\}$. Then, we have $kS^{(i(j))} \geq \sum_{\ell \in B} S^{(\ell)}/2 \geq S/2$, that gives $S/S^{(i(j))} \leq 2k$, as desired.
End of proof of Inequality (29).

Proof of Inequality (30).

First we argue that we can assume that there exists at most one j, with $j \geq k$, with $u_j > S^{(i(j))}/2$. In fact, if there exist $j \neq j'$ such that $u_j > S^{(i(j))}/2$ and $u_{j'} > S^{(i(j'))}/2$ then $i(j') \neq i(j)$ and no element u_r, with $r < k$, is either in bucket $i(j)$ or in $i(j')$. Hence, by the pigeonhole principle, there must exist elements u_r and u_s, with $r, s < k$ that are both in some bucket $i' \notin \{i(j), i(j')\}$. Thus, by Claim 1, swapping buckets for u_r and u_j we get a new set of vectors D' whose buckets are at least as balanced as those of D^* ($H(D') \geq H(D^*)$) and $|D' \cap D| \geq |D^* \cap D|$. However, in D' there is one less index j with $j \geq k$ and $u_j > S^{(i(j))}/2$. Thus, by repeating this argument, we eventually obtain a D' satisfying $H(D^*) = H(D')$ (maximum) and there is at most one j satisfying $u_j > S^{(i(j))}/2$.

We also have that $u_j = u_k$. For otherwise, if $u_k > u_j$, by the previous observation we have that $S^{(i(k))} \geq 2u_k$ hence swapping u_k and u_j we obtain a more balanced set of vectors D' with $H(D') > H(D^*)$, against the hypothesis that $H(D^*)$ is maximum. Therefore, we can assume, w.l.o.g., that $j = k$ and $i(k) = k$.

Finally, for each $\ell, \ell' < k$ we can assume that u_ℓ and $u_{\ell'}$ are in different buckets. For otherwise, swapping buckets for u_j and $u_\ell \geq u_j = u_k > S^{(k)} - u_k$ we get a new set D' with $H(D') \geq H(D^*)$, $|D' \cap D| \geq |D^* \cap D|$ and for all $j \geq k$, $u_j \leq S^{(i(j))}/2$. Then, the desired result would follow because we already proved that inequality (29) holds. Note that $|D' \cap D| \geq |D^* \cap D|$ must hold because the bucket k before the swap cannot be equal to $\{u_k, u_{k+1}, \ldots, u_g\}$ for otherwise we would have an empty bucket.

Because of the previous observations we can assume that in D^*, up to renaming the buckets, for each $m \in [k]$ the element u_m is in bucket m. Let $X_m = S^{(m)} - u_m$. Note that $u_k + \sum_{m=1}^{k} X_m = S$.

Then, we have the following lower bound on the impurity of the buckets of D^*:

$$\sum_{i=1}^{k} I_{Ent}(\mathbf{u}\circ\mathbf{d}_*^{(i)}) \geq \sum_{m=1}^{k} u_m \log\frac{S^{(m)}}{u_m} \geq u_k\left(\sum_{m=1}^{k}\log\frac{S^{(m)}}{u_m}\right) \tag{34}$$

$$= u_k \log\left(\prod_{m=1}^{k}\frac{(u_m+X_m)}{u_m}\right) = u_k\log\left(\frac{(u_k+X_k)\prod_{m=1}^{k-1}(u_m+X_m)}{u_k\prod_{m=1}^{k-1}u_m}\right). \tag{35}$$

On the other hand, because of the standing assumption, $j=k$, we can write as upper bound on the only summand in the left hand side of (30)

$$u_j\log\frac{S}{u_j} = u_k\log\left(\frac{(u_k+X_k)+\sum_{m=1}^{k-1}X_m}{u_k}\right).$$

Therefore, to prove the bound in (30) it is enough to show

$$(u_k+X_k)+\sum_{m=1}^{k-1}X_m \leq \left(\frac{(u_k+X_k)\prod_{m=1}^{k-1}(X_m+u_m)}{\prod_{m=1}^{k-1}u_m}\right).$$

We can now show that this inequality holds by using Claim 2 (ii), which gives $(u_k+X_k)\geq u_s$ for each $s<k$ such that $X_s\neq 0$. Therefore, we have

$$\begin{aligned}
\left((X_k+u_k)+\sum_{s=1}^{k-1}X_s\right)\prod_{m=1}^{k-1}u_m &= (X_k+u_k)\prod_{m=1}^{k-1}u_m+\sum_{s=1}^{k-1}X_s\prod_{m=1}^{k-1}u_m\\
&\leq (X_k+u_k)\prod_{m=1}^{k-1}u_m+\sum_{s=1}^{k-1}\frac{u_k+X_k}{u_s}X_s\prod_{m=1}^{k-1}u_m\\
&= (X_k+u_k)\prod_{m=1}^{k-1}u_m+(u_k+X_k)\sum_{s=1}^{k-1}\left(X_s\prod_{m\in[k-1]\setminus s}u_m\right)\\
&= (u_k+X_k)\left(\prod_{m=1}^{k-1}u_m+\sum_{s=1}^{k-1}X_s\prod_{m\in[k-1]\setminus s}u_m\right)\\
&\leq (u_k+X_k)\prod_{m=1}^{k-1}(u_m+X_m),
\end{aligned}$$

which concludes the proof of (30).

The proof of the lemma is complete. □

By (14), combining the results in the previous lemma with Theorems 3, 4 and the fact that $k\leq n$, we have the following results that apply regardless the relation between g and k.

Theorem 5. *Let (V, I_{Ent}, k) be an instance of* PMWIP *and let* $p=\min\{\log k,\log g\}+\log(\sum_{\mathbf{v}\in V}\|\mathbf{v}\|_1)$. *Then, $\mathcal{A}^{Dom}$ on instance (V, I_{Ent}, k) guarantees $2p$-approximation.*

Theorem 6. *Let Uniform-*PMWIP *(*U-PMWIP*) be the variant of* PMWIP *where all vectors have the same ℓ_1 norm. We have that $\mathcal{A}^{Dom}$ is an $O(\log g+\log n)$-approximation algorithm for* U-PMWIP *with $I=I_{Ent}$.*

5 An $O(\log^2(\min\{g,k\}))$-approximation for PMWIP with I_{Ent}

In this section we present our main result on the entropy measure. Under the assumption $g \le k$, we will show the existence of an $O(\log^2 g)$-approximation polynomial time algorithm. Note that in the light of Lemma 10 and the approach of Section 3 (see, in particular equation (13)), this implies an $O(\log^2(\min\{g,k\}))$-approximation algorithm for any g and k.

Recall that a vector $\mathbf{v}$ is called i-dominant if i is the largest component in $\mathbf{v}$, i.e., $v_i = \|\mathbf{v}\|_\infty$. Accordingly, we say that a set of vectors B (often, in this section, referred to as a bucket) is i-dominant if i is the largest component in the bucket, i.e., $\|\sum_{\mathbf{v}\in B}\mathbf{v}\|_\infty = \sum_{\mathbf{v}\in B} v_i$. We use $dom(\mathbf{v})$ and $dom(B)$, respectively, to denote the index of the dominant component of vectors $\mathbf{v}$ and $\sum_{\mathbf{v}\in B}\mathbf{v}$.

We will say that a bucket B is i-pure if each vector in B is i-dominant. A bucket which is not i-pure for any i will be called a *mixed bucket*.

Following the bound on the impurity of a vector $\mathbf{v}$ given by Lemma 9, we define the *ratio of a vector* $\mathbf{v}$ as

$$ratio(\mathbf{v}) = \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty}.$$

and, accordingly, the *ratio of bucket* B as

$$ratio(B) = \frac{\|\sum_{\mathbf{v}\in B}\mathbf{v}\|_1}{\|\sum_{\mathbf{v}\in B}\mathbf{v}\|_1 - \|\sum_{\mathbf{v}\in B}\mathbf{v}\|_\infty}.$$

Abusing notation, for a set of vectors B we will use $\|B\|_1$ to denote $\|\sum_{\mathbf{v}\in B}\mathbf{v}\|_1$ and $\|B\|_\infty$ to denote $\|\sum_{\mathbf{v}\in B}\mathbf{v}\|_\infty$. Moreover, we use $B(j)$ to denote the set of the j vectors in B of minimum ratio. Since in this section we are only focusing on the entropy impurity measure, we will use I to denote I_{Ent}

We will find it useful to employ the following corollary of Lemma 9.

Corollary 1. *For a vector $\mathbf{v} \in \mathbb{R}^g_+$ and $i \in [g]$ we have*

$$(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)\max\left\{1, \log\left(\frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty}\right)\right\} \le I_{Ent}(\mathbf{v}) \le 2(\|\mathbf{v}\|_1 - v_i)\log\left(\frac{2g\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - v_i}\right) \quad (36)$$

Proof. The second inequality follows from Lemma 9 and Proposition 2, using $A = 2g\|\mathbf{v}\|_1$. □

5.1 Our Tools

In this section we discuss the main tools employed to design our algorithms.

The example of Remark 2, apart from establishing the tightness of $\mathcal{A}^{Dom}$ for I_{Ent}, also shows that we cannot obtain a very good partition by just considering those containing only pure buckets. However, perhaps surprisingly, the situation is different if we allow at most one mixed bucket. This is formalized in Theorem 7, our first and main tool to obtain good approximate solutions for instances of PMWIP. This structural theorem will be used by our algorithms to restrict the space where a partition with low impurity is searched. Its proof, presented in the next section, is reasonably involved: it consists of starting with an optimal partition an then showing how to exchange vectors from its buckets so that a new partition $\mathcal{P}'$ satisfying the desired properties is obtained.

Theorem 7. *There exists a partition $\mathcal{P}'$ with the following properties: (i) it has at most one mixed bucket; (ii) if $\mathbf{v}$ is an i-dominant vector in the mixed bucket and $\mathbf{v}'$ is an i-dominant vector of a i-pure bucket, then $ratio(\mathbf{v}) \leq ratio(\mathbf{v}')$; (iii) the impurity of $\mathcal{P}'$ is at an $O(\log^2 g)$ factor from the minimum possible impurity.*

Our second tool is a transformation χ^{2C} that maps vectors in $\mathbb{R}^g$ into vectors in $\mathbb{R}^2$. The nice property of this transformation is that it preserves the entropy of a set of i-pure vectors up to an $O(\log g)$ distortion as formalized by Proposition 3. Thus, in the light of Theorem 7, instead of searching for low-impurity partitions of g-dimensional vectors with at least k-1 pure buckets, we can search for those in a 2-dimensional space.

The transformation χ^{2C} is defined as follows

$$\chi^{2C}(\mathbf{v}) = \begin{cases} (\|\mathbf{v}\|_\infty, \|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) & \text{if } \|\mathbf{v}\|_\infty \geq \frac{1}{2}\|\mathbf{v}\|_1 \\ (\|\mathbf{v}\|_1/2, \|\mathbf{v}\|_1/2) & \text{if } \|\mathbf{v}\|_\infty < \frac{1}{2}\|\mathbf{v}\|_1. \end{cases}$$

Let $I_2(B)$ to denote the *2-impurity of the set* B, that is, the impurity of the set of 2-dimensional vectors obtained by applying χ^{2C} to each vector in B. We have that

Proposition 3. *Fix $i \in [g]$ and let B be an i-pure bucket. It holds that*

$$(1/2)I_2(B) \leq I(B) \leq 2I_2(B) + 4(\log g)\sum_{\mathbf{w}\in B} I(\mathbf{w}).$$

Finally, our last tool is the following result from [18], here stated following our notation, that shows that PMWIP can be optimally solved when $g = 2$.

Theorem 8 ([18])**.** *Let V be a set of 2-dimensional vectors and let k be an integer larger than 1. There exists a polynomial time algorithm to build a partition of V into k buckets with optimal impurity.*

In addition, the partition computed by the algorithm satisfies the following property: if B is a bucket in the partition and if $\mathbf{v} \in V \setminus B$ then either $ratio(\mathbf{v}) \geq \max_{\mathbf{v}'\in B}\{ratio(\mathbf{v}')\}$ or $ratio(\mathbf{v}) \leq \min_{\mathbf{v}'\in B}\{ratio(\mathbf{v}')\}$.

Motivated by the previous results we define $\mathcal{A}^{2C}$ as the algorithm that takes as input a set of vectors B and an integer b and produces a partition of B into b buckets by executing the following steps: (i) every vector $\mathbf{v} \in B$ is mapped to $\chi^{2C}(\mathbf{v})$; (ii) the algorithm given by Theorem 8 is applied over the transformed set of vectors to distribute them into b buckets; (iii) the partition of B corresponding to the partition produced in step (ii) is returned.

Algorithm $\mathcal{A}^{2C}$ is employed as a subroutine of the algorithms presented in the next section. The following property holds for $\mathcal{A}^{2C}$.

Proposition 4. *Let B be an i-pure set of vectors. The impurity of the partition $\mathcal{P}$ constructed by the algorithm $\mathcal{A}^{2C}$ on input (B, b) is at most an $O(\log g)$ factor from the minimum possible impurity for a partition of set B into b buckets.*

Proof. Let $\mathcal{P}^*$ be the partition of B into b buckets with minimum impurity. We have that

$$\begin{aligned} I(\mathcal{P}) &\leq 2I_2(\mathcal{P}) + 4\log g\sum_{\mathbf{w}\in B} I(\mathbf{w}) \leq 2I_2(\mathcal{P}^*) + 4\log g\sum_{\mathbf{w}\in B} I(\mathbf{w}) \\ &\leq 4I(\mathcal{P}^*) + 4(\log g)\sum_{\mathbf{w}\in B} I(\mathbf{w}) = O(\log g)I(\mathcal{P}^*), \end{aligned}$$

where the first inequality follows from Proposition 3 (applied to each bucket of $\mathcal{P}$), the second one from the optimality of $\mathcal{P}$, the third one by Proposition 3 (applied to each bucket of $\mathcal{P}^*$), and the last one by observing that by superadditivity of I we have $I(\mathcal{P}^*) \geq \sum_{\mathbf{w} \in B} I(\mathbf{w})$. $\square$

5.2 Proof of Theorem 7

The proof proceeds in steps. Lemma 11 shows that there exists a partition with at most one mixed bucket whose impurity is $O(\log g)$ factor from $\mathrm{OPT}(V)$. Next, we explain how to modify this partition in order to obtain a new partition $\mathcal{P}$ with at most one mixed bucket, impurity limited by $O(\log g)\mathrm{OPT}(V)$ and such that the vectors in its i-pure buckets are ordered according to their ratios. Finally, we show how to modify $\mathcal{P}$ so that we obtain a partition $\mathcal{P}'$ that satisfies the properties of Theorem 7.

Lemma 11. *There exists a partition with at most one mixed bucket that satisfies: (i) the impurity of the mixed bucket is at a $O(\log g)$ factor from the optimal impurity and (ii) the sum of the impurities of the pure buckets is at most the optimal impurity.*

Proof. Let $\mathcal{P}^*$ be an optimal partition. If $\mathcal{P}^*$ has at most one mixed bucket we are done. Otherwise, let $B_1, \ldots, B_j$, with $j \geq 2$, be the mixed buckets in $\mathcal{P}^*$. We assume w.l.o.g. that B_1 is the bucket with the smallest ratio among the mixed buckets.

For $i = 2, \ldots, j$, let $S_i = \{\mathbf{v} | \mathbf{v} \in B_i \text{ and } dom(\mathbf{v}) \neq dom(B_i)\}$. Let $\mathcal{P}$ be a new partition obtained from $\mathcal{P}^*$ by replacing B_1 with $B_1' = B_1 \cup S_2 \ldots \cup S_j$ and B_i with $B_i' = B_i \setminus S_i$, for $i \geq 2$. It is clear that B_1' is the unique mixed bucket in $\mathcal{P}$.

It follows from subadditivity that $I(B_i') \leq I(B_i)$ for $i > 1$, which establishes (ii). Thus, in order to complete the proof it is enough to establish an upper bound on $I(B_1')$.

For $i = 2, \ldots, j$, let $\mathbf{u}^{(i)} = \sum_{\mathbf{v} \in B_i} \mathbf{v}$ and $\mathbf{w}^{(i)} = \sum_{\mathbf{v} \in S_i} \mathbf{v}$. Moreover, let $s_i = \|\mathbf{w}^{(i)}\|_1$. Thus,

$$\|\mathbf{u}^{(i)}\|_1 - \|\mathbf{u}^{(i)}\|_\infty = \|\mathbf{u}^{(i)}\|_1 - \sum_{\mathbf{v} \in B_i} v_{dom(B_i)} \geq |\mathbf{w}^{(i)}\|_1 - \sum_{\mathbf{v} \in S_i} v_{dom(B_i)} \geq \|\mathbf{w}^{(i)}\|_1/2 = \frac{s_i}{2},$$

where the leftmost inequality holds because for each $\mathbf{v} \in S_i$ we have $dom(\mathbf{v}) \neq dom(B_i)$, so that $\|\mathbf{v}\|_1/2 \geq v_{dom(B_i)}$.

Therefore, it follows from Corollary 1 that

$$I(B_i) \geq (\|\mathbf{u}^{(i)}\|_1 - \|\mathbf{u}^{(i)}\|_\infty) \max\{1, \log(ratio(B_i))\} \geq \frac{s_i}{2} \max\{1, \log(ratio(B_i))\}, \tag{37}$$

for each $i > 1$

We assume w.l.o.g. that B_1 is 1-dominant. Let $\mathbf{u}^{(1)} = \sum_{\mathbf{v} \in B_1} \mathbf{v}$ and let $s_1 = \|\mathbf{u}^{(1)}\|_1$ and $c_1 = \|\mathbf{u}^{(1)}\|_1 - \|\mathbf{u}^{(1)}\|_\infty$. Again, from Corollary 1, we have

$$I(B_1) \geq c_1 \max\{1, \log(ratio(B_1))\} \tag{38}$$

For $i = 2, \ldots, j$ let $c_i = \|\mathbf{w}^{(i)}\|_1 - w_1^{(i)}$. Let $\mathbf{u} = \sum_{\mathbf{v} \in B_1'} \mathbf{v}$. Then, $\mathbf{u} = \mathbf{u}^{(1)} + \sum_{i=2}^{j} \mathbf{w}^{(i)}$, hence $\|\mathbf{u}\|_1 = \sum_{i=1}^{j} s_i$ and $\|\mathbf{u}\|_1 - u_1 = \sum_{i=1}^{j} c_i$.

By Corollary 9 (with $i = 1$) we have that

$$
\begin{aligned}
I(B'_1) &\leq 2\left(\sum_{i=1}^{j} c_i\right) \cdot \log\left(2g\frac{\sum_{i=1}^{j} s_i}{\sum_{i=1}^{j} c_i}\right) \\
&\leq 2\left(c_1 + \sum_{i=2}^{j} s_i\right) \log\left(2g\frac{s_1 + \sum_{i=2}^{j} s_i}{c_1 + \sum_{i=2}^{j} s_i}\right) && (39) \\
&\leq 2\left(c_1 + \sum_{i=2}^{j} s_i\right) \log\left(2g\frac{s_1}{c_1}\right) = 2\left(c_1 + \sum_{i=2}^{j} s_i\right) \log(2g \cdot ratio(B_1)), && (40)
\end{aligned}
$$

where the second inequality follows from Proposition 2.

Since $ratio(B_1) \leq ratio(B_i)$ for $i > 1$ we can conclude, by using the lower bounds (37) and (38) that $I(B'_1) = O(\log g) \sum_{i=1}^{j} I(B_i)$. □

Using the mapping χ^{2C} and Proposition 4, we can derive the following result.

Lemma 12. *There exists a partition with the following properties: (i) it has at most one mixed bucket; (ii) if B_i is a i-pure bucket and $\mathbf{v}$ is a i-dominant vector that belongs to an i-pure bucket different from B_i then either $ratio(\mathbf{v}) \geq \max_{\mathbf{v}' \in B_i}\{ratio(\mathbf{v}')\}$ or $ratio(\mathbf{v}) \leq \min_{\mathbf{v}' \in B_i}\{ratio(\mathbf{v}')\}$ and (iii) its impurity is at a $O(\log g)$ factor from the minimum possible impurity.*

Proof. Let $\mathcal{P}$ be a partition that satisfies Lemma 11. Let V_i be the set of i-dominant vectors that are not in the mixed bucket. If $V_i \neq \emptyset$ let $B_i^1, \ldots, B_i^{t(i)}$ be the i-pure buckets where they lie. We replace these $t(i)$ buckets by the $t(i)$ buckets obtained by running algorithm $\mathcal{A}^{2C}$ for input $(V_i, t(i))$. This replacement is applied for every i. It follows from Proposition 4 that the total impurity of the pure buckets in the new partition is at most at a $O(\log g)$ factor from the total impurity of the pure buckets in $\mathcal{P}$.

The property (ii) is assured by the structure of the partition constructed by Algorithm $\mathcal{A}^{2C}$. In order to guarantee that the ties are broken correctly we present the i-dominant vector for algorithm $\mathcal{A}^{2C}$ in the order of their ratios. □

Now, we conclude the proof of Theorem 7. Our starting point is the partition $\mathcal{P}$ that satisfies items (i)-(iii) of Lemma 12. We show how to obtain a partition $\mathcal{P}'$ from $\mathcal{P}$ that satisfies the properties of Theorem 7.

Let B_{mix} be the mixed bucket in $\mathcal{P}$. We assume w.l.o.g that $dom(B_{mix}) = 1$. Moreover, let B_i be the i-pure bucket that contains the i-dominant vectors with the smallest ratios. In what follows we assume that the vectors in B_i are sorted by increasing order of their ratios so that by the jth first vector in B_i we mean the one with the jth smallest ratio.

Let $s_{i,p} = \|B_i\|_1$ (p indicates a pure bucket, i indicates the dominance, and s indicates that we are considering the total sum of the components of the vectors). Let $V_{i,mix}$ be the set of i-dominant vectors in B_{mix}, i.e., $V_{i,mix} = \{\mathbf{v} \in B_{mix} \mid dom(\mathbf{v}) = i\}$.

Let $s_{i,mix} = \|V_{i,mix}\|_1$, i.e., $s_{i,mix}$ denotes the total sum of the components of the i-dominant vectors from bucket B_{mix}.

In order to explain the construction of $\mathcal{P}'$ we need to define $2g$ set of vectors $X_1, Y_1, \ldots, X_g, Y_g$ that will be moved among the buckets of $\mathcal{P}$ to obtain $\mathcal{P}'$. Those are defined according to the following cases:

case 1. $s_{i,p} < s_{i,mix}$.

subcase 1.1 $i > 1$ (i is not the dominant component of B_{mix}).

Let r_i be the largest ratio among the ratios of the vectors from B_i. In addition, let $Y_i = B_i$ and let X_i be the set of i-dominant vectors from B_{mix} whose ratios are larger than r_i.

subcase 1.2 $i = 1$. Let m be such that $\|V_{1,mix}(m-1)\|_1 \leq s_{1,p}$ and $\|V_{1,mix}(m)\|_1 > s_{1,p}$.

Moreover, let r_1 be the ratio of the m-th first 1-dominant vector of $V_{1,mix}$. Let $X_1 = V_{1,mix} \setminus V_{1,mix}(m-1)$ (the set containing all the 1-dominant vector of B_{mix} but the first $m-1$ ones) and let $Y_1 = \{\mathbf{v} \in B_1 \mid \mathit{ratio}(\mathbf{v}) < r_1\}$ (the set containing every vector in B_1 with ratio smaller than r_1).

case 2. $s_{i,p} \geq s_{i,mix}$.

In this case, let m be such that $\|B_i(m-1)\|_1 < s_{i,mix}$ and $\|B_i(m)\|_1 \geq s_{i,mix}$. Moreover, let r_i be the ratio of the m-th vector of B_i. We define $Y_i = B_i(m-1)$ (the set containing the $m-1$ first vectors of B_i) and $X_i = \{\mathbf{v} \in V_{i,mix} \mid \mathit{ratio}(\mathbf{v}) > r_i\}$ (the set containing every i-dominant vector in B_{mix} with ratio larger than r_i).

Let $X = \bigcup_{i=1}^{g} X_i$ and let $Y = \bigcup_{i=1}^{g} Y_i$. The partition $\mathcal{P}'$ is obtained from $\mathcal{P}$ by replacing the bucket B_{mix} with the bucket $B'_{mix} = (B_{mix} \cup Y) \setminus X$ and the bucket B_i, for every i, with $B'_i = (B_i \cup X_i) \setminus Y_i$.

Lemma 13. *The partition $\mathcal{P}'$ satisfies item (i) and (ii) from Theorem 7.*

Proof. By construction every i-dominant vector in B'_{mix} has ratio at most r_i and every i-dominant vector in $V \setminus B'_{mix}$ has ratio at least r_i. □

Lemma 14. *The impurity of the partition $\mathcal{P}'$ is at most $O(\log g)$ times larger than that of $\mathcal{P}$.*

Proof. See the Appendix. □

5.3 The approximation algorithm

We first present a pseudo-polynomial time algorithm that provides an $O(\log^2 g)$ approximation and then we show how to convert it into a polynomial time algorithm with the same approximation. The key idea is to look among the partitions that satisfy the properties of Theorem 7 for one that (roughly speaking) minimizes the impurity of its mixed bucket plus the sum of the 2-impurity of its pure buckets.

A special case: no mixed bucket. Theorem 7 establishes the existence of a partition $\mathcal{P}^*$ whose impurity is an $O(\log^2 g)$ approximation of the optimum and has at most one mixed bucket. For a better understanding of the strategy at the basis of our algorithm, let us first discuss how one can efficiently construct a good partition for the case where the partition $\mathcal{P}^*$, achieving the $O(\log^2 g)$ approximation, has no mixed buckets.

In this case, we can employ algorithm $\mathcal{A}^{2C}$ to obtain a partition with minimum 2-impurity among those that only have pure buckets. By Proposition 3 it follows that the impurity of a partition made only of pure buckets, is upper bounded by its 2-impurity plus $O(\log g)$ times a lower bound on the optimal impurity. Proceeding like in the proof of Proposition 4 then we can show that the impurity of the partition of minimum 2-impurity is upper bounded by the same upper bound on the impurity of $\mathcal{P}^*$.

The partition with minimum 2-impurity made only of pure buckets can be obtained by means of dynamic programming.

To see this, for each $j = 1, \ldots, g$ let

$$V_j = \{\mathbf{v} | dom(\mathbf{v}) = j\} \quad \text{and} \quad S_j = \{\mathbf{v} | dom(\mathbf{v}) = j' \text{ for some } j' \leq j\} \tag{41}$$

Moreover, for each $b = 1, \ldots, k$ let $Q^*(S_j, b)$ be a partition of the vectors of S_j into b pure buckets such that its 2-impurity, denoted by $\mathrm{OPT}_2(j, b)$, is minimum. It is not hard to see that the following recurrence holds:

$$\mathrm{OPT}_2(j, b) = \begin{cases} I_2(\mathcal{A}^{2C}(V_j, b)) & \text{if } j = 1 \\ \min_{1 \leq b' < b - j}\{I_2(\mathcal{A}^{2C}(V_j, b')) + \mathrm{OPT}_2(j-1, b-b')\} & \text{if } j > 1 \end{cases} \tag{42}$$

where $\mathcal{A}^{2C}(V_j, b)$ is the partition of V_j into b buckets obtained by the Algorithm $\mathcal{A}^{2C}$ discussed in the previous section.

Thus, if there exists a partition $\mathcal{P}^*$, without mixed buckets, for which $I(\mathcal{P}^*) = O(\log^2 g)\mathrm{OPT}(V)$, then the impurity of the partition $\mathcal{Q}^*(S_g, k)$ constructed by a DP algorithm based on the equation (42) satisfies

$$\begin{aligned} I(\mathcal{Q}^*(S_g, k)) \;&\leq\; I_2(\mathcal{Q}^*(S_g, k)) + O(\log g) \sum_{\mathbf{v} \in V} I(\mathbf{v}) \\ &\leq\; I_2(\mathcal{P}^*) + O(\log g)\mathrm{OPT}(V) \leq 2I(\mathcal{P}^*) + O(\log g)\mathrm{OPT}(V) \leq O(\log^2 g)OPT(V), \end{aligned}$$

where the first inequality in the first line follows from Proposition 3, the first inequality in the second line is due to the minimality of the 2-impurity of $\mathcal{Q}^*(S_g, k)$ and the superadditivity of I impying that $\sum_{\mathbf{v} \in V} I(\mathbf{v})$ is a lower bound on $\mathrm{OPT}(V)$.

A pseudopolynomial time algorithm for the general case. Now, we turn to the case where there exists at most one mixed bucket in the partition given by Theorem 7. Given an instance (V, I, k) of PMWIP, let $C = \sum_{\mathbf{v} \in V} \|\mathbf{v}\|_1$ and for each $i = 1, \ldots, g$, let V_i and S_i be as in (41). For fixed $w, i \in [g], \ell \in [|V|], c \in [C], b \in [k]$ let us denote by $\mathcal{Q}^*(w, \ell, S_i, b, c)$ a partition of S_i into b buckets that satisfies the following properties:

- a it has one bucket, denoted by $B^{\mathcal{Q}^*}$, that contains exactly ℓ vectors that are w-dominant;
- b it contains at most one mixed bucket. This mixed bucket, if it exists, is the bucket $B^{\mathcal{Q}^*}$.
- c For every i, if $\mathbf{v}$ and $\mathbf{v}'$ are, respectively, i-dominant vectors in $B^{\mathcal{Q}^*}$ and $V \setminus B^{\mathcal{Q}^*}$; then $ratio(\mathbf{v}) \leq ratio(\mathbf{v}')$;
- d the total sum of all but the w-component of vectors in $B^{\mathcal{Q}^*}$ is equal to c, i.e., $c = \|B^{\mathcal{Q}^*}\|_1 - (\sum_{\mathbf{v} \in B^{\mathcal{Q}^*}} v_w)$;
- e the sum of the 2-impurities of the buckets in $\mathcal{Q}^*(w, \ell, S_j, b, c) \setminus B^{\mathcal{Q}^*}$ is minimum among the partitions for S_j into b buckets that satisfy the previous items.

The algorithm builds partitions $\mathcal{Q}^* = \mathcal{Q}^*(w, \ell, S_g, k, c)$ for all possible combinations of w, ℓ and c and, then, returns the one with minimum impurity.

This approach is motivated by the following: Let $\mathcal{P}^*$ be a partition that contains one mixed bucket, denoted by B^*_{mix}, and satisfies the properties of Theorem 7. For such a partition, let $w^* = dom(B^*_{mix})$, ℓ^* be the number of w^*-dominant vectors in B^*_{mix} and $c^* = \|B^*_{mix}\|_1 - \sum_{\mathbf{v}\in B^*_{mix}} v_{w^*}$ (the sum of all but the w^* component of the vectors in B^*_{mix}.) Then, it is possible to prove that the impurity of a partition $\mathcal{Q}^* = \mathcal{Q}^*(w^*, \ell^*, S_g, k, c^*)$ is at an $O(\log g)$ factor from that of $\mathcal{P}^*$ (see the proof of Theorem 9 below). The key observations are: (i) the impurity of the bucket $B^{\mathcal{Q}^*}$ of $\mathcal{Q}^*$ is at an $O(\log g)$ factor from that of B^*_{mix} since $\|B^{\mathcal{Q}^*}\|_1$ is at most twice $\|B^*_{mix}\|_1$ and $\|B^{\mathcal{Q}^*}\|_1 - \sum_{\mathbf{v}\in B^{\mathcal{Q}^*}} v_{w^*} = \|B^*_{mix}\|_1 - \sum_{\mathbf{v}\in B^*_{mix}} v_{w^*} = c^*$; (ii) the sum of the 2-impurity of the buckets in $\mathcal{Q}^* \setminus B^{\mathcal{Q}^*}$ is at most the sum of the 2-impurity of the buckets $\mathcal{P}^* \setminus B^*_{mix}$ so that their standard impurities differ by not more than a logarithmic factor.

Building the partitions $\mathcal{Q}^*(w, \ell, S_i, b, c)$. To simplify our discussion let us assume w.l.o.g. that $w = 1$.

Let $\mathcal{Q}^* = \mathcal{Q}^*(w, \ell, S_i, b, c)$ be a partition that satisfies properties (a)-(e) above and let $I_2^{pure}(\mathcal{Q}^*) = I_2(\mathcal{Q}^* \setminus B^{\mathcal{Q}^*})$ be the total 2-impurity of the buckets of $\mathcal{Q}$ which are surely pure. Moreover, let $V_i(j)$ be the set of the j vectors of V_i of smallest ratio, and let $c_i(j) = \|V_i(j)\|_1 - \sum_{\mathbf{v}\in V_i(j)} v_1$, i.e., the total sum of all components but the first of the vectors in $V_i(j)$.

For $i = 1$ we have

$$I_2^{pure}(\mathcal{Q}^*(1, \ell, S_i, b, c)) = \begin{cases} I_2(\mathcal{A}^{2C}(V_1 \setminus V_1(\ell), b-1)), & \text{if } c = c_1(\ell) \\ \infty & \text{otherwise} \end{cases} \tag{43}$$

For $i > 1$ we have

$$I_2^{pure}(\mathcal{Q}^*(1, \ell, S_i, b, c)) = \min_{\substack{0\le j\le |V_i| \\ 0\le b' < b}} \{I_2(\mathcal{A}^{2C}(V_i \setminus V_i(j), b')) + I_2^{pure}(\mathcal{Q}^*(1, \ell, S_{i-1}, b-b', c-c_i(j)))\} \tag{44}$$

Algorithm 1 relies on equations (43) and (44). First, at line 1, it preprocesses the partitions generated by algorithm $\mathcal{A}^{2C}$ that are used by these equations. Next, it runs over the possible combinations (w, ℓ) and, for each of them, the procedure $\mathcal{M}$ is called to search for a partition with impurity smaller than those found so far.

For a fixed pair (w, ℓ), procedure $\mathcal{M}$ constructs partitions $\mathcal{Q}^*(w, \ell, S_i, b, c)$ for all the possible combinations of i and b and all the possible corresponding c. Thus, to simplify we use $\mathcal{Q}^*(S_i, b, c)$ to refer to $\mathcal{Q}^*(w, \ell, S_i, b, c)$. The first step of procedure $\mathcal{M}$, where component w is relabeled to 1 is only meant to keep a direct correspondence with the assumption $w = 1$ in equations (43) and (44). Equation (43) is implemented in lines 8-10 to build the list U_1 that contains all the partitions $\mathcal{Q}^*(1, b, c)$ for which $I_2^{pure}(\mathcal{Q}^*(1, b, c)) \neq \infty$. The loop of lines 11-12 calls procedure `GenerateNewList`, that employs Equation (44), to build a list U_i, from list U_{i-1}, containing all partitions $\mathcal{Q}^*(i, b, c)$ with $I_2^{pure}() \neq \infty$. We note that at line 20 the special bucket B' of the new partition under construction, is obtained as an extension of the bucket $B^{\mathcal{Q}^*}$ of the partition $\mathcal{Q}^*(i-1, b,)$ in U_{i-1}, which includes the vectors in $V_1(\ell)$.

At the end of the procedure $\mathcal{M}$ the partition of minimum impurity in U_g is returned. This is the partition of minimum impurity among the partition $\mathcal{Q}^*(w, \ell, S_g, b, c)$ stored in list U_g for some b and c. Hence, for $w = w^*$ and $\ell = \ell^*$, in particular, it is a partition that has impurity not larger than the partition $\mathcal{Q}^*(w^*, \ell^*, S_g, k, c^*)$ which we already observed to be an $O(\log g)$ approximation of the minimum impurity partition satisfying Theorem 7.

Since $c \le C = \sum_{\mathbf{v}\in V} \|\mathbf{v}\|_1$ and the lists U_i cannot grow larger than gkC it is easy to see that the proposed algorithm runs in polynomial time on $n = |V|$ and $C = \sum_{\mathbf{v}\in V} \|\mathbf{v}\|_1$.

The following theorem gives a formal proof of the approximation guarantee for the solution returned by Algorithm 1

Algorithm 1 (V: set of g-dimensional vectors; k: integer)

Preprocess $\mathcal{A}^{2C}(V_j \setminus V_j(j'), b)$ for $j = 1, \ldots, g$, $j' = 1, \ldots, |V_j|$ and $b = 1, \ldots, k$
$\mathcal{Q}_{Best} \leftarrow$ arbitrarily chosen partition of S_g into k buckets
for $w = 1, \ldots, g$ and $\ell = 1, \ldots, |V_w|$ **do**
    if $I(\mathcal{M}(w, \ell)) < I(\mathcal{Q}_{Best})$ **then**
        Update $\mathcal{Q}_{Best}$ to $\mathcal{M}(w, \ell)$
procedure $\mathcal{M}$(w:class,ℓ:integer)
    Relabel the components of the vectors so that label of component w becomes 1.
    for $b' = 1, \ldots, k$ **do**
        $\mathcal{Q} \leftarrow \{V_1(\ell)\} \cup \mathcal{A}^{2C}(V_1 \setminus V_1(\ell), b' - 1)$
        Add $\mathcal{Q}$ to U_1.
    for $i = 2, \ldots, g$ **do**
        $U_i \leftarrow$ `GenerateNewList`(U_{i-1})
    Return the partition with minimum impurity in U_g
function `GenerateNewList`(U, i)
    for every partition $\mathcal{Q}$ in the list U **do**
        Let (i, b, c) be the values s.t. $\mathcal{Q} = \mathcal{Q}(i, b, c)$
        if $b < k$ **then**
            for $b' = 1, \ldots, k - b$ **do**
                for $j = 0, \ldots, |V_i|$ **do**
                    $B' \leftarrow B^{\mathcal{Q}} \cup V_i(j)$
                    $\mathcal{Q}' \leftarrow \{B'\} \cup (\mathcal{Q} \setminus B^{\mathcal{Q}}) \cup \mathcal{A}^{2C}(V_i \setminus V_i(j), b')$.
                    Add $\mathcal{Q}'$ to U
                    $c' \leftarrow \|B'\|_1 - \sum_{\mathbf{v}\in B'} v_1$
                    if U contains another $\mathcal{Q}''$ with parameters $(i, b + b', c')$ **then**
                        if $I_2^{pure}(\mathcal{Q}'') > I_2^{pure}(\mathcal{Q}')$ **then**
                            remove $\mathcal{Q}''$ from U
                        else
                            remove $\mathcal{Q}'$ from U
    return U

Theorem 9. *For instances with vectors of dimension $g \le k$, there exists a pseudo-polynomial time $O(\log^2 g)$-approximation algorithm for* PMWIP.

Proof. Let $\mathcal{Q}$ be the partition with smallest impurity between the one returned by Algorithm 1 and the one returned by the DP based algorithm that implements Equation (42). In addition, let $\mathcal{P}^*$ be a partition that satisfies the conditions of Theorem 7. In particular, we have $I(\mathcal{P}^*) \le O(\log^2 g)\mathrm{OPT}(V)$.

To show that $I(\mathcal{Q})$ is $O(\log^2 g)\mathrm{OPT}(V)$ we compare $I(\mathcal{Q})$ with $I(\mathcal{P}^*)$. We argue according to whether $\mathcal{P}^*$ has a mixed bucket or not.

Case 1. $\mathcal{P}^*$ has a mixed bucket. We can assume that $\mathcal{P}^*$ coincides with the partition $\mathcal{P}'$ of Lemma 14.

Let B'_{mix} be the mixed bucket of $\mathcal{P}'$ and assume w.l.o.g. that w' is the dominant component in B'_{mix}. Let $s' = \|B'_{mix}\|_1$, $c' = s' - \|B'_{mix}\|_\infty$ and $c = s' - \sum_{\mathbf{v}\in B'_{mix}} v_1$ (recall that in the proof of Lemma 14 component 1 is the dominant component of the bucket B_{mix} from the partition $\mathcal{P}$ that is used as a basis to obtain $\mathcal{P}'$; note that it is possible to have $1 \neq w'$). From Proposition 2, since $c' \leq c$, and the proof of Lemma 14 we have that

$$2c' \log \frac{2gs'}{c'} \leq 2c \log \frac{2gs'}{c} \leq O(\log^2 g)\mathrm{OPT}(V). \tag{45}$$

In particular, the second inequality in (45) is proved in Appendix D, **Bounds on the mixed buckets** B'_{mix}—note that with our present definition of s' and c the middle term of (45) coincides with the right hand side of (91).

Let ℓ' be the number of w'-dominant vectors from $\mathcal{P}'$ that lie in B'_{mix}. We know that the impurity of the output partition $\mathcal{Q}$ is not larger than that of $\mathcal{Q}^*(w', \ell', S_g, k, c')$, one of the partitions built by Algorithm 1. Thus, it is enough to show that the impurity of $\mathcal{Q}^*(w', \ell', S_g, k, c')$ is at a $O(\log^2 g)$ factor from the optimum. For this we will show that $I(\mathcal{Q}^*)$ is $O(I(\mathcal{P}') + \mathrm{OPT}(V)\log g)$. In what follows we use $\mathcal{Q}^*$ to refer to $\mathcal{Q}^*(w', \ell', S_g, k, c')$, and as before, $B^{\mathcal{Q}^*}$ denotes the special bucket in $\mathcal{Q}^*$.

Let

$$s_1 = \|\sum_{\substack{\mathbf{v}\in B'_{mix} \\ dom(\mathbf{v})=w'}} \mathbf{v}\|_1 \quad \text{and} \quad c_1 = s_1 - \sum_{\substack{\mathbf{v}\in B'_{mix} \\ dom(\mathbf{v})=w'}} v_w.$$

By Corollary 1, with $i = w'$, we have that

$$I(B^{\mathcal{Q}^*}) \leq 2c' \log\left(\frac{2g\cdot(2(c'-c_1)+s_1)}{c'}\right) \leq 2c' \log \frac{4gs'}{c'} \leq O(\log^2 g)\mathrm{OPT}(V) \tag{46}$$

where

- for the first inequality, we are also using the fact that $\|B^{\mathcal{Q}^*}\|_1 \leq 2(c'-c_1)+s_1$. To see that the last relation holds we note that

  $$\|\sum_{\substack{\mathbf{v}\in B^{\mathcal{Q}^*} \\ dom(\mathbf{v})\neq w'}} \mathbf{v}\|_1 - \sum_{\substack{\mathbf{v}\in B^{\mathcal{Q}^*} \\ dom(\mathbf{v})\neq w'}} v_{w'} = c' - c_1,$$

  hence $2(c'-c_1)$ is an upper bound on the total mass of the vectors in $B^{\mathcal{Q}^*}$ which are not w'-dominant. Therefore, we have the upper bound $2(c'-c_1)+s_1$ used in the first inequality for $\|B^{\mathcal{Q}^*}\|_1$.

- for the second inequality we are using $s' \geq c' - c_1 + s_1$.

- the last inequality follows from (45)

We now focus on the buckets of $\mathcal{Q}^*$ different from $B^{\mathcal{Q}^*}$—which are surely pure. From the proof of Lemma 14 (Appendix D, **Bounds on the i-pure buckets**) we have that the total impurity of the buckets in $\mathcal{P}'$ different from B'_{mix} satisfies

$$\sum_{B\in\mathcal{P}'\setminus B'_{mix}} I(B) = O(\log^2 g)\mathrm{OPT}(V). \tag{47}$$

In addition, we have

$$\begin{aligned}
\sum_{B\in\mathcal{Q}^*\setminus B^{\mathcal{Q}^*}} I(B) &\leq \sum_{B\in\mathcal{Q}^*\setminus B^{\mathcal{Q}^*}} \big(2I_2(B) + 4(\log g)\sum_{\mathbf{w}\in B} I(\mathbf{w})\big) && (48)\\
&= 2\sum_{B\in\mathcal{Q}^*\setminus B^{\mathcal{Q}^*}} I_2(B) + 4(\log g)\sum_{\mathbf{w}\in V} I(\mathbf{w}) && (49)\\
&\leq 2\sum_{B\in\mathcal{P}'\setminus B'_{mix}} I_2(B) + 4(\log g)\sum_{\mathbf{w}\in V} I(\mathbf{w}) && (50)\\
&\leq 4\sum_{B\in\mathcal{P}'\setminus B'_{mix}} I(B) + 4(\log g)\sum_{\mathbf{w}\in V} I(\mathbf{w}) && (51)\\
&\leq O(\log^2)\mathrm{OPT}(V) + \mathrm{OPT}(V)\log g, && (52)
\end{aligned}$$

where the inequality in (48) follows from Proposition 3; (50) follows from (49) by the property (e); (51) follows from (50) by Proposition 3 and, finally, to obtain (52) we use (47) for the left term and superadditivity for the right term;

From (46) and (48)-(51) we have

$$I(\mathcal{Q}) \leq I(\mathcal{Q}^*) = I(B^{\mathcal{Q}^*}) + \sum_{B\in\mathcal{Q}^*\setminus B^{\mathcal{Q}^*}} I(B) = O(\log^2 g)\mathrm{OPT}(V)$$

and the proof for *Case 1* is complete.

Case 2. $\mathcal{P}^*$ does not have a mixed bucket. In this case, let $\mathcal{Q}'$ be the partition built according to the recurrence in (42). It was argued right after this inequality that $I(\mathcal{Q}')$ is $O(\log^2 g)OPT(V)$. Thus, $I(\mathcal{Q})$ is also $O(\log^2 g)OPT(V)$. $\square$

The polynomial time algorithm. Let $\mathcal{P}^*$ be a partition that satisfies the conditions of Theorem 7. If $\mathcal{P}^*$ does not have a mixed bucket then the DP based on Equation (42) is a polynomial time algorithm that builds a partition whose impurity is at most $O(\log g)$ times larger than that of $\mathcal{P}^*$. Thus, we just need to focus in the case where $\mathcal{P}^*$ has a mixed bucket.

Let Algo-Prune be the variant of Algorithm 1 that together with the instance takes as input an extra integer parameter t and uses the following additional conditions regarding the way the lists $U_i's$ are handled: (i) only partitions for which the fifth parameter c is at most t are added to U_i; (ii) after creating the list U_i in line 12 and before proceeding to list U_{i+1} the following pruning is performed: the interval $[0,t]$ is split into $4g$ subintervals of length $t/4g$ and while there exist two partitions $\mathcal{Q}(w,\ell,S_i,b,c)$ and $\mathcal{Q}'(w,\ell,S_i,b,c')$ in U_i with both c' and c lying in the same subinterval, the one for which the $I_2^{pure}()$ is larger is removed. This step guarantees that a polynomial number of partitions are kept in U_i.

Let us consider the algorithm $\mathcal{A}_{poly}$ that executes Algo-Prune $e = \lceil\log(\sum_{\mathbf{v}\in V}\|\mathbf{v}\|_1)\rceil$ times. In the jth execution Algo-Prune is called with $t = 2^j$. After execution j the partition with the minimum impurity found in U_g is kept as $\mathcal{Q}^{(j)}$. After all the e executions have been performed, the partition with minimum impurity in $\{\mathcal{Q}^{(1)},\ldots,\mathcal{Q}^{(e)}\}$ is returned.

From the above observation that in each call of Algo-Prune the number of partitions kept in the lists is polynomial in size of the instance and the fact that the number of calls to Algo-Prune is also polynomial in the size of the input, we have that $\mathcal{A}_{poly}$ is a polynomial time algorithm for our problem.

It remains to show that $\mathcal{A}_{poly}$ is also an $O(\log^3 g)$-approximation algorithm. For this, let us consider again the partitions $\mathcal{P}^*$ and $\mathcal{Q}^*(1, \ell^*, S_g, k, c)$ defined in the case 2 of the proof of Theorem 9. We can show that there is a partition $\mathcal{Q}$ among those constructed by $\mathcal{A}_{poly}$ such that $I_2^{pure}(\mathcal{Q}) \leq I_2^{pure}(\mathcal{Q}^*(1, \ell^*, S_g, k, c))$ and such that the special bucket $B^{\mathcal{Q}}$ of $\mathcal{Q}$ has ℓ^* vectors that are 1-dominant and satisfies $\|B^{\mathcal{Q}}\|_1 - \sum_{\mathbf{v} \in B^{\mathcal{Q}}} v_1 \leq 2(\|B_{mix}\|_1 - \|B_{mix}\|_\infty) = 2c$.

Note that these properties are enough to obtain our claim since, with them, proceeding as in the proof of Theorem 9 one can show that the impurity of $\mathcal{Q}$ is at most an $O(\log^3 g)$ factor larger than the optimal impurity.

For the definition of $\mathcal{Q}$ we need some additional notation. As in Theorem 9, let us denote with $\mathcal{Q}^*$ the partition $\mathcal{Q}^*(1, \ell^*, S_g, k, c)$. Then $B^{\mathcal{Q}^*}$ denotes the special bucket of this partition.

For $i = 1, \ldots, g$ let b_i be the number of i-pure buckets in $\mathcal{Q}^* \setminus B^{\mathcal{Q}^*}$ and let n_i be the number of i-dominant vectors that lie in the bucket $B^{\mathcal{Q}^*}$. Moreover, let $c_i = \|V_i(n_i)\|_1 - \sum_{\mathbf{v} \in V_i(n_i)} v_1$. With this, we have that $\sum_{i=1}^{g} c_i = c = \|B_{mix}\|_1 - \|B_{mix}\|_\infty$.

The partition $\mathcal{Q}$ is defined as the last partition of the sequence $\mathcal{Q}_1, \ldots, \mathcal{Q}_g$, where

- $\mathcal{Q}_1$ is the partition $\mathcal{Q}^*(1, \ell^*, S_1, b_1, c_1)$ constructed in the $\lceil \log c \rceil$-th execution of Algo-Prune, i.e., with $t = 2^{\lceil \log c \rceil} > c$.
- For $i > 1$, let $\mathcal{Q}_i'$ be the partition obtained by extending $\mathcal{Q}_{i-1}$ with the b_i buckets from the partition $\mathcal{A}^{2C}(V_i \setminus V_i(n_i), b_i)$ and replacing the bucket $B^{\mathcal{Q}_{i-1}}$, from $\mathcal{Q}_{i-1}$, with $B^{\mathcal{Q}_{i-1}} \cup V_i(n_i)$. Note that such a partition is added to U_i before the pruning step (ii) is executed. Then, $\mathcal{Q}_i$ is defined as the partition that survives (after the pruning step (ii)) in the subinterval where $\mathcal{Q}_i'$ lies.

Let $c_i' = \|B^{\mathcal{Q}_i}\|_1 - \sum_{\mathbf{v} \in B^{\mathcal{Q}_i}} v_1$ (the total mass of vectors in the special bucket $B^{\mathcal{Q}_i}$ of $\mathcal{Q}_i$, minus the mass of such vectors in the component 1).

We can prove by induction that

$$\left| c_i' - \sum_{j=1}^{i} c_j \right| \leq \frac{i \cdot t}{4g}.$$

For $i = 1$ the result holds since $c_1 = c_1'$. It follows from the induction that

$$c_{i-1}' - \sum_{j=1}^{i-1} c_j \leq \frac{(i-1) \cdot t}{4g}$$

The result for i is established by observing that the pruning step (ii) above, ensures that

$$\left| c_i' - (c_{i-1}' + c_i) \right| \leq \frac{t}{4g}$$

Let $\mathcal{Q}_i^*$ be the subpartition of $\mathcal{Q}^*$ that contains bucket $B^{\mathcal{Q}^*}$ and all i'-pure bucket for each $i' \leq i$. We can also prove by induction that $I_2^{pure}(\mathcal{Q}_i) \leq I_2^{pure}(\mathcal{Q}_i^*)$. For $i = 1$ the result holds since $\mathcal{Q}_1 = \mathcal{Q}_1^*$. For a general i we have that

$$\begin{aligned} I_2^{pure}(\mathcal{Q}_i) &\leq I_2^{pure}(\mathcal{Q}_{i-1}) + \mathcal{A}^{2C}(V_i \setminus V(n_i), b_i) \\ &\leq I_2^{pure}(\mathcal{Q}_{i-1}^*) + \mathcal{A}^{2C}(V_i \setminus V(n_i), b_i) = I_2^{pure}(\mathcal{Q}_i^*). \end{aligned}$$

Thus, by using the same arguments employed in the proof of Theorem 9 on can show that the impurity of $\mathcal{Q}$ is at an $O(\log^2 g)$ factor from the optimal one. We can now state the main theorem of the paper.

Theorem 10. *There is a polynomial time $O(\log^2(\min\{g, k\}))$ approximation algorithm for* PMWIP.

Proof. By the above argument we have that Algorithm $\mathcal{A}_{poly}$ is a polynomial time $O(\log^2 g)$ approximation algorithm for PMWIP with $g \leq k$. For $g > k$, applying Lemma 10 and the approach of Section 3 (see, in particular equation (13)), we have an $O(\log^2 k)$-approximation algorithm. Putting together the two cases we have the claim. □

6 Hardness of Approximation of PMWIP$_{Ent}$

The goal of this section is to establish the following result.

Theorem 11. PMWIP$_{Ent}$ *is APX-Hard.*

We start with the definition of a gap decision problem associated with the minimum vertex cover problem on bounded degree graphs.

Definition 1. *For every $\epsilon > 0$ and integer* D*, we define the following (gap) decision problem: ϵ-*GAP-MINVC-3*: given a cubic graphs $G = (V, E)$ and an integer k, decide whether G has a vertex cover of size k or all vertex covers of G have size $> k(1 + \epsilon)$.*

Definition 2. *For every $\eta > 0$, we define the following (gap) decision problem: η-*GAP-PMWIP$_{Ent}$*: given a set of vectors U, an integer k, and a value k', decide whether there exists a k-clustering $\mathcal{C} = C_1, \ldots C_k$ of the vectors in U such that the total impurity $I_{Ent}(\mathcal{C}) = \sum_{\ell=1}^{k} I_{Ent}(C_\ell)$ is at most k' or for each k-clustering $\mathcal{C}$ of U it holds that $I_{Ent}(\mathcal{C}) > (1 + \eta)k'$.*

We will use the following result from the proof of [10, Theorems 17 and 19] as the basis for a gap-preserving reduction from MINIMUM VERTEX COVER IN CUBIC GRAPHS to PMWIP$_{Ent}$, which will in turn imply APX-hardness of the latter problem.

Theorem 12. *For some constants $\epsilon > 0$ the ϵ-*GAP-MINVC-3 *is NP-hard.*

We will use the following definition of a (constant) gap-preserving reduction:

Definition 3. *Let $\mathbb{A}, \mathbb{B}$ be minimization problems. A gap-preserving reduction from $\mathbb{A}$ to $\mathbb{B}$ is a polynomial time algorithm that, given an instance x of $\mathbb{A}$ and a value k, produces an instance y of $\mathbb{B}$ and a value k' such that there exists constants $\epsilon, \eta > 0$ for which*

1. *if $OPT(x) \leq k$ then $OPT(y) \leq k'$;*
2. *if $OPT(x) > (1 + \epsilon)k$ then $OPT(y) > (1 + \eta)k'$;*

The reduction. Given a cubic graph $G = (V = \{v_1, \ldots, v_n\}, E)$, with $|E| = m$, we can construct (in polynomial time) a set of m n-dimensional binary vectors $U = \{v^e \mid e \in E\} \subseteq \{0, 1\}^n$ by stipulating that if $e = (v_i, v_j)$ then only the i-th and j-th components of v^e are 1 and all others are 0.

In what follows, for a set of vectors $C \subseteq U$ we use $I_{Ent}(C)$ to denote the impurity of C, that is, $I_{Ent}(C) = I_{Ent}(\sum_{\mathbf{v} \in C} \mathbf{v})$.

We will find it convenient to visualize vectors in U in terms of their corresponding edges. For a subset of $C \subseteq U$ we will say that C is a *p-star* if the corresponding set of edges form a star in G, i.e., if $|C| = p$ and there exists a coordinate $j \in [n]$ such that for each vectors $v^e \in C$ the jth components of v^e is 1. By directly applying the definition of the impurity I_{Ent} we have the following.

Fact 2. *If $C \subseteq U$ is a p-star, then we have $I_{Ent}(C) = 2p + p \log p$*

We also have that p-stars are sets of p edges of minimum impurity as recorded in the following fact.

Fact 3. *Let $C \subseteq U$ be a set of p edges. Then $I_{Ent}(C) \geq 2p + p \log p$.*

Proof. Let $d^C(v)$ be the number of edges of C incident in v. Then, we have

$$I_{Ent}(C) = 2pH\left(\frac{d^C(v_1)}{2p}, \ldots, \frac{d^C(v_n)}{2p}\right), \tag{53}$$

where $H()$ denotes the Shannon entropy.

By the concavity of $H()$, we have that the minimum of the entropy appearing on rights hand side of (53) is attained when the maximum possible mass is concentrate in one component, i.e.,

$$\min_{\substack{d_1, \ldots d_n \\ d_i \in [p], \sum_i d_i = 2p}} H\left(\frac{d_1}{2p}, \ldots, \frac{d_n}{2p}\right) = H\left(\frac{p}{2p}, \frac{1}{2p}, \ldots \frac{1}{2p}, 0, \ldots, 0\right). \tag{54}$$

The desired result now follows by noticing that the entropy on the right hand side of (54) is equal to $1 + \frac{1}{2} \log p$. $\square$

The following lemma, which is key for our development, relates minimal vertex covers with star decompositions in cubic graphs. This result might be of independent interest.

Lemma 15. *Let $G = (V, E)$ be a cubic graph and U be a corresponding set of vectors obtained by the reduction described above. If G has a minimal vertex cover of size k then there is a k-clustering $\mathcal{C} = \{C_1, \ldots C_k\}$ of U where for each i, the cluster C_i is either a 2-star or a 3-star.*

Proof. Let $S = \{v_1, \ldots, v_k\}$ be a minimal vertex cover in G. Let $G[S] = (S, E[S])$ be the subgraph of G induced by the vertices in S.

For each $i = 1, \ldots, k$ let D_i be the set of edges in $E \setminus E[S]$ that are incident to v_i.

We say that v_i is of type j if $|D_i| = j$. Since the graph is cubic and S is minimal then the only possible types are 1, 2 and 3. We then extend the sets D_i by applying the following procedure:

> **While** there exists an edge $e = v_i v_j \in G[S]$, with v_i having type 2 **do**
> Remove e from $G[S]$ and add it to D_j
> Increase the type of v_j by one unit.

Let $G[S]$ be the resulting graph. Since the graph G was cubic, every vertex of type 2 or 3 in $G[S]$ is isolated and every vertex of type 1 is adjacent to two vertices of type 1. Therefore, in $G[S]$, the edges define a collection of disjoint cycles of vertices of type 1. Let $v_{i_1}, v_{i_2}, \ldots, v_{i_r}$ be one such cycle, where $v_{i_r} = v_{i_1}$. Then, for $j = 1, \dots, r-1$ we remove edge $(v_{i_j}, v_{i_{j+1}})$ from $G[S]$ and add it to D_{i_j} so that v_{i_j} becomes a vertex of type 2. After repeating this for each cycle, we have that for all j, the set D_j has cardinality 2 or 3. Moreover, by construction, we also have $\cup_i D_i = E$.

Then, the clustering $\mathcal{C} = \{C_1, \dots, C_k\}$ where $C_i = \{v^e \mid e \in D_i\}$, satisfies the claim. □

Corollary 2. *Let $G = (V, E)$ be a cubic graph and U be a corresponding set of vectors obtained by the reduction described above. If G has a minimal vertex cover of size k then there is a k-clustering $\mathcal{C} = \{C_1, \dots C_k\}$ of U of impurity $I_{Ent}(\mathcal{C}) = 6k + 3(|U| - 2k)\log 3$.*

Proof. Let $m = |U| = |E|$. Let $\mathcal{C} = \{C_1, \dots C_k\}$ be the clustering given by Lemma 15. Let a and b denote the number of 2 and 3 stars in $\mathcal{C}$, respectively. Then $2a + 3b = m$ and $a + b = k$, whence, $a = 3k - m$ and $b = m - 2k$. Finally, from Fact 2 we have

$$I_{Ent}(\mathcal{C}) = 6a + b\,(6 + 3\log 3) = 6k + 3(m - 2k)\log 3.$$

□

The consequence of the last corollary is that setting $k' = 6k + 3(m - 2k)\log 3$ we have that the reduction described above satisfies property 1 in Definition 3.

6.1 Preservation of the Gap

We want to show that when the minimum vertex cover of the graph G has size at least $k(1 + \epsilon)$ then the impurity of every k-clustering is at least a constant times larger than $6k + 3(|U| - 2k)\log 3$, which is the impurity of the clustering in Corollary 2, which exists when the minimum vertex cover of G has cardinality $\leq k$. This will imply that our reduction satisfies also the second property in Definition 3.

In the following, $\mathcal{C}$ will denote a clustering of minimum impurity for the instance of PMWIP_{Ent} obtained via the reduction, when, for some constant $\epsilon > 0$, the size of the minimum vertex cover for G is at least $k(1 + \epsilon)$.

We will use the following notation to describe such a clustering $\mathcal{C}$ of minimum impurity.

- a: number of clusters in $\mathcal{C}$ consisting of a 3-star; we refer to these clusters as the a-group of clusters;
- b: number of clusters in $\mathcal{C}$ consisting of a 2-star; we refer to these clusters as the b-group of clusters;
- c: number of cluster in $\mathcal{C}$ consisting of a 1-star (single edge); we refer to these clusters as the c-group of clusters;
- d: number of clusters in $\mathcal{C}$ consisting of 2 edges without common vertex (2-matching); we refer to these clusters as the d-group of clusters;
- e: number of remaining clusters in $\mathcal{C}$; we refer to these clusters as the e-group of clusters;
- q: number of edges in the e-group of clusters.

In the definitions above the letters a, b, c, d and e are used to denote both the size and the type of a group of clusters. We believe this overloaded notation helps the readability.

The following proposition will be useful in our analysis.

Proposition 5. *Let $x \geq 2$ and let n_1,n_2 be positive integers. We have that*

$$n_1(2x + x \log x) + n_2(2(x+1) + (x+1)\log(x+1)) \geq (n_1+n_2)(2\overline{x} + \overline{x}\log\overline{x}),$$

where $\overline{x} = (n_1 x + n_2(x+1))/(n_1+n_2)$.

Proof. It is enough to prove that

$$n_1 x \log x + n_2(x+1)\log(x+1) \geq (n_1+n_2)\overline{x}\log\overline{x}.$$

This inequality follows from Jensen inequality since $f(x) = x \log x$ is convex in the interval $[2, \infty]$ $\square$

The next two propositions give lower bounds on the sum of the impurities of the clusters in the e-group.

Proposition 6. *The total impurity of the clusters in the e-group is at least $2q + q/e \log(q/e)$.*

Proof. Let $C_1, \ldots, C_e$ be the clusters in the e-group. Note that each one of these clusters has cardinality ≥ 3. Let $p = \lfloor q/e \rfloor$. Then, we have $p \geq 3$. Suppose that there exist clusters C_i, C_j such that $|C_i| = x$ and $|C_j| = y$ with $y > x+1$. Let C_i' be an $(x+1)$-star and C_j' be a $(y-1)$-star, then we have

$$\begin{aligned} I_{Ent}(C_i) + I_{Ent}(C_j) &\geq 2x + x\log x + 2y + y \log y \\ &\geq 2(x+1) + (x+1)\log(x+1) + 2(y-1) + (y-1)\log(y-1) \\ &\geq I_{Ent}(C_i') + I_{Ent}(C_j'), \end{aligned}$$

where the first inequality follows from Fact 3, the second inequality holds true for each $3 \leq x \leq y-2$ and the last inequality follows from Fact 2.

The above inequality says that if we replace C_i, C_j with C_i', C_j', the impurity of the resulting set of e clusters is not larger than the impurity of the original e-group. Moreover, the total number of edges has not changed. By repeated application of such a replacement we eventually obtain a group of e clusters $\tilde{\mathcal{C}} = \{\tilde{C}_1, \ldots, \tilde{C}_e\}$ each of cardinality p or $p+1$ and containing in total e edges and such that $\sum_{i=1} I_{Ent}(C_i) \geq \sum_{i=1} I_{Ent}(\tilde{C}_i)$. In particular, the total impurity of such clusters is not larger than the total impurity of the original e clusters. Note that these new e clusters need not exist and are only used here for the sake of the analysis.

Let n_1 be the number of clusters in $\tilde{\mathcal{C}}$ with p edges and n_2 be the number of clusters in $\tilde{\mathcal{C}}$ with $p+1$ edges and let $\overline{p} = \frac{n_1 p + n_2(p+1)}{n_1+n_2}$. Then, $q = n_1 p + n_2(p+1)$ and $e = n_1 + n_2$, hence $\overline{p} = q/e$.

Finally, by applying Proposition 5, we have the desired result:

$$\begin{aligned} \sum_{j=1}^{e} I_{Ent}(C_i) &\geq \sum_{j=1}^{e} I_{Ent}(\tilde{C}_i) \\ &= n_1(2p + p\log p) + n_2(2(p+1) + (p+1)\log(p+1)) \\ &\geq (n_1+n_2)(2\overline{p} + \overline{p}\log\overline{p}) = 2q + \frac{q}{e}\log\frac{q}{e}. \end{aligned}$$

$\square$

We say that C is an *S-structure* if it correspond to a set of 3 edges, of which two form a 2-star and the third one is not incident to any one of the first two edges. Moreover, we say that C is a 3-*path structure* if it corresponds to a set of 3 edges forming a path.

Fact 4. *The impurity of a 3-path structure is* $2+6\log 3$*, the impurity of a S-structure is* $4+6\log 3$ *and the impurity of a cluster corresponding to 3 disjoint edges is* $6+6\log 3$.

Proposition 7. *If $q < 4e$ then the total impurity of the clusters in the e-group is lower bounded by the impurity of a partition of q edges into e sets, each of which is either a 4-star or a 3-path structure.*

Proof. First we consider two simple cases. In the first one no cluster in the e-group has cardinality larger than 3. In this case, all clusters have cardinality 3 and the result holds since Fact 4 assures that the impurity of each of these clusters can be lower bounded by the impurity of a $3-$path cluster. In the second case, all clusters in the e-group, with more than three edges, are 4 stars. Again, we can establish the result by using the same reasoning.

In the most interesting case there is a cluster C in the e-group with cardinality $r \geq 4$ that does not correspond to a $4-$star. In this case, there exist $r-3$ clusters of cardinality 3 in the e-group, for otherwise the average cardinality would be ≥ 4, violating the assumption $q < 4e$. Let $D_1, \ldots D_{r-3}$ be these clusters. Moreover, let C' be a 3-path structure and $D'_1, \ldots D'_{r-3}$ be 4-stars. We have

$$|C'| + \sum_{j=1}^{r-3} |D'_j| = 3 + 4(r-3) = r + 3(r-3) = |C| + \sum_{j=1}^{r-3} |D_j|. \tag{55}$$

Furthermore, by definition clusters of cardinality 3 that are in the e-group are not 3-stars. Thus, by Fact 4 we have that for each $i = 1, \ldots r-3$, it holds that $I_{Ent}(D_i) \geq 2 + 6\log 3$ and by Fact 3, it holds that $I_{Ent}(C) \geq 2r + r\log r$. Then

$$\begin{aligned} I_{Ent}(C) + \sum_{j=1}^{r-3} I_{Ent}(D_j) &\geq 2r + r\log r + (r-3)\,(2 + 6\log 3) \\ &\geq (2 + 6\log 3) + 16(r-3) \\ &= I_{Ent}(C') + \sum_{i=1}^{r-3} I_{Ent}(D'_i), \end{aligned}$$

where the second inequality holds for each $r \geq 4$.

The above inequalities together with (55) say that if we replace $C, D_1, \ldots D_{r-3}$ with $C', D'_1, \ldots D'_{r-3}$, the impurity of the resulting set of e clusters is not larger than the impurity of the original set of e clusters. Moreover, the total number of edges does not change. By repeated application of such a replacement we eventually obtain a group of e clusters $\tilde{\mathcal{C}} = \{\tilde{C}_1, \ldots, \tilde{C}_e\}$ each of which is either a 3-path structure of a 4-star. Moreover, these clusters contain in total e edges and $\sum_{i=1} I_{Ent}(C_i) \geq \sum_{i=1} I_{Ent}(\tilde{C}_i)$, i.e., the total impurity of the new clusters[1] is not larger than the total impurity of the original e clusters. The proof is complete. □

Proposition 8. *If the minimum vertex cover for G has size at least $k(1+\epsilon)$ then the following inequality holds: $c + d + q \geq k\epsilon/2$.*

[1]Note that these e clusters need not exist and are only used here for the sake of the analysis.

Proof. If it does not hold we could construct a vertex cover of size smaller than $k(1+\epsilon)$ by selecting a vertex to cover the edges of the 3-stars, b vertices to cover the edges of the b stars and one vertex per edge of the other clusters. The number of edges in these other clusters is $c+2d+q$. Hence, we must have $a+b+c+2d+q \geq k(1+\epsilon)$. Since $a+b \leq k$ we conclude that $c+d+2q \geq k\epsilon$, so that $c+d+q \geq c/2+d/2+q \geq k\epsilon/2$ □

Let $\mathcal{C}^{(k)}$ denote the clustering only consisting of 2-stars and 3-stars described in Lemma 15 and Corollary 2, which exists when the minimum vertex cover of the graph G has size $\leq k$. We also refer to this clustering as the *k-cover clustering*.

We will now show that, if the minimum size of a vertex cover for G is at least $k(1+\epsilon)$ then the impurity of the minimum impurity clustering $\mathcal{C}$ (for the instance obtained via the reduction) is at least a constant factor larger than the impurity of $\mathcal{C}^{(k)}$.

Lemma 16. *Let $G=(V,E)$ be a cubic graph and U be a corresponding set of vectors obtained by the reduction described above. If every vertex cover in G has size $\geq k(1+\epsilon)$, then there exists a constant $\eta > 0$ such that every k-clustering $\mathcal{C}=\{C_1,\dots C_k\}$ of U has impurity $I_{Ent}(\mathcal{C}) \geq k'(1+\eta)$, where $k' = I_{Ent}(\mathcal{C}^{(k)})$.*

Proof. Recall the definition of the parameters a,b,c,d,e,q, regarding the minimum impurity k-clustering $\mathcal{C}$. We split our analysis into two cases according whether $3k-m$, which is the number of 2-stars in $\mathcal{C}^{(k)}$, is smaller than $b+d$, the number of clusters of $\mathcal{C}$ with 2 edges, or not.

Case 1 $b+d \geq 3k-m$.

Let $z=(b+d)-(3k-m)$. We can write the impurity of the k-cover clustering $\mathcal{C}^{(k)}$ as

$$I_{Ent}(\mathcal{C}^{(k)}) = (6+3\log 3)a + (6+3\log 3)z + 6(3k-m).$$

Let $\overline{p} = q/e$. Then, lower bounding the impurity of the clusters in the e-group as in Proposition 6, we have that the impurity of $\mathcal{C}$ satisfies

$$I_{Ent}(\mathcal{C}) \geq (6+3\log 3)a + 2c + 8d + 6b + (\overline{p}\log\overline{p} + 2\overline{p})e.$$

Therefore,

$$I_{Ent}(\mathcal{C}) - I_{Ent}(\mathcal{C}^{(k)}) \geq 2c + 2d - (3\log 3)z + (\overline{p}\log\overline{p} + 2\overline{p})e$$

Summing up the edges in the clusters, we have that $2z+\overline{p}e+c = 3(z+c+e)$, hence $z=(\overline{p}-3)e-2c$ Thus,

$$\begin{aligned}
I_{Ent}(\mathcal{C}) - I_{Ent}(\mathcal{C}^{(k)}) &\geq (2+6\log 3)c + 2d + (\overline{p}\log\overline{p} + (2-3\log 3)\overline{p} + 9\log 3)e && (56)\\
&= (2+6\log 3)c + 2d + (\log\overline{p} + (2-3\log 3) + 9\log 3/\overline{p})q && (57)\\
&> 1.99(c+d+q) && (58)\\
&\geq 0.49k\epsilon, && (59)
\end{aligned}$$

where inequality (57)-(58) follows because the global minimum of $\log\overline{p} + (2-3\log 3) + 9\log 3/\overline{p}$ is larger than 1.99 and inequality (58)-(59) follows from Proposition 8.

Since $I_{Ent}(\mathcal{C}^{(k)}) \leq k(6+3\log 3)$, the impurity of the clustering $\mathcal{C}$ is a constant factor larger than the impurity of $\mathcal{C}^{(k)}$.

Case 2 $b+d < 3k-m$. We further split this case into two subcases according to whether the average size of the clusters in the e-group is at least four or smaller than four.

Subcase 2.1 $b + d < 3k - m$ and $q \geq 4e$.

Let $y = (m - 2k - a)$ and $z = 3k - m - (b + d)$. Let $\overline{p} = q/e$. Hence, we have $\overline{p} \geq 4$.

We can write the impurity of the k-cover clustering $\mathcal{C}^{(k)}$ as

$$I_{Ent}(\mathcal{C}^{(k)}) = (6 + 3\log 3)a + (6 + 3\log 3)y + 6b + 6d + 6z.$$

Since $y + z = (c + e)$ and $3y + 2z = \overline{p}e + c$ we get that $y = (\overline{p} - 2)e - c$ and $z = 2c + (3 - \overline{p})e$. Thus,

$$\begin{aligned} I_{Ent}(\mathcal{C}^{(k)}) &= (6 + 3\log 3)a + (\overline{p} - 2)(6 + 3\log 3)e - (6 + 3\log 3)c + 6b + 6d + 12c + 6(3 - \overline{p})e \\ &= (6 + 3\log 3)a + [(3\log 3)\overline{p} + 6(1 - \log 3)]e + 6b + 6d + (6 - 3\log 3)c. \end{aligned}$$

The impurity of $\mathcal{C}$ is given by

$$I_{Ent}(\mathcal{C}) \geq (6 + 3\log 3)a + 6b + 2c + 8d + (\overline{p}\log \overline{p} + 2\overline{p})e$$

Therefore, we have

$$\begin{aligned} I_{Ent}(\mathcal{C}) - I_{Ent}(\mathcal{C}^{(k)}) &\geq (3\log 3 - 4)c + 2d + [\overline{p}\log \overline{p} + (2 - 3\log 3)\overline{p} + 6(\log 3 - 1)]e && (60) \\ &= (3\log 3 - 4)c + 2d + (\log \overline{p} + (2 - 3\log 3) + 6(\log 3 - 1)/\overline{p})q && (61) \\ &\geq (2.5 - 1.5\log 3)(c + d + q) && (62) \\ &\geq (1.25 - 0.75\log 3)k\epsilon && (63) \end{aligned}$$

where the inequality (61) holds because function $f(x) = \log x + 6(\log 3 - 1)/x$ is increasing in the interval $[4, \infty]$. The last inequality follows from Proposition 8.

Since $I_{Ent}(\mathcal{C}^{(k)}) \leq k(6 + 3\log 3)$, the impurity of the clustering $\mathcal{C}$ is at least constant factor larger than the impurity of $\mathcal{C}^{(k)}$.

Subcase 2.2 $b + d < 3k - m$ and $q < 4e$.

Let $z = (3k - m) - (b + d)$ and $y = (m - 2k) - a$. Because of the assumption $q < 4e$, by Proposition 7, the sum of the impurities of the clusters in the e-group can be lower bounded by the sum of the impurities of a set of e clusters such that each of them is either a 3-path structure or a 4-star. Let x be the number of 3-path structures in this group, hence $(e - x)$ is the number of 4-stars in this same group.

Since both $\mathcal{C}^{(k)}$ and $\mathcal{C}$ have the same number of clusters, we have

$$c + e = z + y.$$

Since the total number of edges in the clusters is the same we also have

$$c + 3x + 4(e - x) = 2z + 3y.$$

From these equalities we have that $z = 2c - (e - x)$.

Therefore, we can write the impurity of the clustering $\mathcal{C}^{(k)}$ as follows

$$I_{Ent}(\mathcal{C}^{(k)}) = 6(b + d) + 6z + (6 + 3\log 3)(a + c + e - z).$$

On the other hand, for the impurity of $\mathcal{C}$ we have

$$I_{Ent}(\mathcal{C}) \geq 6b + 8d + a(6 + 3\log 3) + 2c + 16(e - x) + (2 + 6\log 3)x.$$

Therefore, we have

$$\begin{aligned}
I_{Ent}(\mathcal{C}) - I_{Ent}(\mathcal{C}^{(k)}) &\geq 2d + 2c - (6 + 3\log 3)(c + e - z) - 6z + 16(e - x) + (2 + 6\log 3)x && (64)\\
&= 2d - (4 + 3\log 3)c - (6 + 3\log 3)e + (3\log 3)z + (2 + 6\log 3)x \\
&\qquad\qquad +16(e - x) && (65)\\
&= 2d + (3\log 3 - 4)c - (6 + 6\log 3)e + (2 + 9\log 3)x + 16(e - x) && (66)\\
&= 2d + (3\log 3 - 4)c + (10 - 6\log 3)e + (9\log 3 - 14)x && (67)\\
&\geq 2d + (3\log 3 - 4)c + (10 - 6\log 3)q/4 \geq \frac{10 - 6\log 3}{4}(d + c + q) && (68)\\
&\geq \frac{10 - 6\log 3}{8}k\epsilon, && (69)
\end{aligned}$$

where (66) follows from (65) using $z = 2c - e + x$, the last but first inequality holds by $q < 4e$ and the last inequality follows from Proposition 8.

Since $I_{Ent}(\mathcal{C}^{(k)}) \leq k(6+3\log 3)$, the impurity of the clustering $\mathcal{C}$ is a constant factor larger than the impurity of the k-cover cluster $\mathcal{C}^{(k)}$.

As a result of the above case based analysis, setting

$$\eta = \epsilon \times (6 + 3\log 3) \times \min\left\{0.49, \frac{5 - 3\log 3}{4}, \frac{10 - 6\log 3}{8}\right\},$$

we have $I_{Ent}(\mathcal{C}) \geq I_{Ent}(\mathcal{C}^{(k)})(1 + \eta)$, as desired.

□

Proof of Theorem 11. We have shown that for every $\epsilon > 0$ there exists an $\eta > 0$ such that for every instance $(G = (V, E), k)$ of ϵ-GAP-MINVC-3, setting $k' = 6k + 3(|E| - 2k)\log 3$ the instance (U, k, k') of η-GAP-PMWIP$_{Ent}$ produced according to our reduction is such that if G has a vertex cover of size $\leq k$ then U has a k-clustering of impurity $\leq k'$ and if all vertex covers of G have size $> (1 + \epsilon)k$ then all k-clustering of U have impurity $> (1 + \eta)k'$.

This, together with Theorem 12 implies that there exists $\eta > 0$ such that the η-GAP-PMWIP$_{Ent}$ is NP-hard. Hence, if $P \neq NP$ there is no polynomial time $(1 + \eta)$-approximation algorithm for PMWIP$_{Ent}$. □

6.2 The impact of our result on related problems

PMWIP$_{Ent}$ is closely related to the MTC_{KL} [9], the problem of clustering a set of n probability distributions into k groups minimizing the total Kullback-Leibler (KL) divergence from the distributions to the centroids of their assigned groups. Mathematically, we are given a set of n points $p^{(1)}, \ldots, p^{(n)}$, corresponding to probability distributions, and a positive integer k. The goal is to find a partition of the points into k groups $V_1, \ldots, V_k$ and a centroid $c^{(i)}$ for each group V_i such that

$$\sum_{i=1}^{k} \sum_{p \in V_i} KL(p, c^{(i)})$$

is minimized, where $KL(p,q) = \sum_{j=1}^{d} p_j \ln(p_j/q_j)$ is the Kullback-Leibler divergence between points p and q.

It is known that in the optimal solution for each $i = 1, \dots k$ the centroid $c^{(i)} = (c_1^{(i)}, \dots c_d^{(i)})$ is given by $c_j^{(i)} = \sum_{p \in V_i} p_j / |V_i|$, for each $j = 1, \dots, d$. Thus, MTC_{KL} is equivalent to the problem of finding a partition that minimizes

$$\sum_{i=1}^{k} \sum_{p \in V_i} KL(p, c^{(i)}) = \sum_{i=1}^{k} \sum_{p \in V_i} \sum_{j=1}^{d} p_j (\ln p_j - \ln c_j^{(i)}) = \tag{70}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{d} p_j^{(i)} \ln p_j^{(i)} - \sum_{i=1}^{k} \sum_{p \in V_i} \sum_{j=1}^{d} p_j \ln c_j^{(i)} = \tag{71}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{d} p_j^{(i)} \ln p_j^{(i)} - \sum_{i=1}^{k} \sum_{j=1}^{d} \left(\sum_{p \in V_i} p_j \right) \ln \frac{\left(\sum_{p \in V_i} p_j \right)}{|V_i|} = \tag{72}$$

$$-\frac{1}{\log e} \sum_{i=1}^{n} I_{Ent}(p^{(i)}) + \frac{1}{\log e} \sum_{i=1}^{k} I_{Ent}\left(\sum_{p \in V_i} p \right) \tag{73}$$

Therefore, the optimal solution of MTC_{KL} is equal to the optimal one of the particular case of PMWIP$_{Ent}$ in which $\mathbf{v}_i = p_i$ for $i = 1, \dots, n$. While their optimal solutions match in this case, PMWIP$_{Ent}$ and MTC_{KL} differ in terms of approximation since the objective function for MTC_{KL} has an additional constant term $-\sum_{i=1}^{n} I_{Ent}(p^{(i)})$ so that an α-approximation for MCT_{KL} problem implies an α-approximation for PMWIP$_{Ent}$ while the converse is not necessarily true.

In terms of computational complexity, Chaudhuri and McGregor [9] proved that the variant of MTC_{KL} where the centroids must be chosen from the input probability distributions is NP-Complete. The NP-Hardness of MTC_{KL}, that remained open in [9], was established in Ackermann et. al. [2], where it is also mentioned that the APX-hardness of k-means in $\mathbb{R}^2$ would imply the same kind of hardness for MTC_{KL}. However, it is not known whether the former is APX-Hard.

Our result provides an important progress in this line of investigation since it implies the APX-hardness of MTC_{KL}. This follows from the above observation about the correspondence between the two problems and the fact that the same arguments in our reduction can also be used to show the inapproximability of instances where all vectors have ℓ_1 norm equal to any constant value, and in particular 1, i.e., the case where PMWIP$_{Ent}$ corresponds to MTC$_{KL}$. In summary we have the following.

Corollary 3. MCT$_{KL}$ *is APX-Hard.*

7 New fast method for information clustering

We have designed, implemented and tested a novel heuristic Ratio-Greedy for clustering based on Entropy impurity, which uses some of the ideas that lead to the algorithms presented in the previous sections.

The basis of Ratio-Greedy is the dominance algorithm of Section 4. In the case where the number of allowed cluster k is larger than the dimension g, the dominance algorithm would only use g clusters, each one being a pure bucket. Ratio-Greedy after grouping the vectors into $g < k$

pure buckets $B_1, \ldots, B_g$, according to the dominant component, proceeds with splitting for each $i = 1, \ldots, g$, the bucket B_i into t_i clusters, so that $\sum_i t_i = k$, i.e., eventually producing exactly k clusters. The number of clusters t_i in which B_i is divided is chosen to be proportional to the impurity of B_i relative to the total impurity of $B_1, \ldots, B_g$, i.e., $t_i = \min\{|B_i|, k\frac{I_{Ent}(B_i)}{\sum_i I_{Ent}(B_i)}\}$. For partitioning B_i into t_i clusters, Ratio-Greedy uses ideas from the analysis of Section 5: It constructs a partition $B_{i,1}, \ldots, B_{i,t_i}$ where for each j and each vector $\mathbf{v} \notin B_{i,j}$ either $ratio(\mathbf{v}) \geq \max_{\mathbf{v}' \in B_{i,j}}\{ratio(\mathbf{v}')\}$ or $ratio(\mathbf{v}) \leq \min_{\mathbf{v}' \in B_{i,j}}\{ratio(\mathbf{v}')\}$. This is achieved in the following way: Let L_i be the list obtained by sorting the vectors in B_i according to their *ratios* as defined in Section 5. It will be convenient to think of L_i as list of $|L_i|$ singleton clusters: $\{\mathbf{v}_1\}, \{\mathbf{v}_2\}, \ldots, \{\mathbf{v}_{n_i}\}$, where n_i denotes the number of vectors in B_i. A greedy approach is used to reduce the number of clusters in L_i from n_i to t_i. This procedure consists of iteratively selecting two adjacent clusters $\{\mathbf{v}_i, \mathbf{v}_{i+1} \ldots, \mathbf{v}_j\}$ and $\{\mathbf{v}_{j+1}, \mathbf{v}_{j+2} \ldots, \mathbf{v}_\ell\}$ in the current list L_i and replacing them with their union $\{\mathbf{v}_i, \mathbf{v}_{i+1} \ldots, \mathbf{v}_\ell\}$ so that a new list containing one less cluster is obtained. The pair of adjacent clusters in the list L_i that is selected to be merged, at each iteration, is the one for which $loss(\cdot, \cdot)$ is minimum, where the $loss(C, C')$ of two clusters C and C' is given by $loss(C, C') = I_{Ent}(C \cup C') - I_{Ent}(C) - I_{Ent}(C')$. The procedure stops when the list L_i contains exactly t_i clusters.

We need time $O(n_i \cdot g + n_i \log n_i)$ to: (i) compute the ratio of each vector in B_i; (ii) sort them; (iii) compute the impurity of each vector; (iv) compute for each pair of adjacent vectors in L_i the loss of merging them into one cluster.

We maintain a priority queue whose elements are the pair of adjacent clusters (C_j, C_{j+1}) in L_i (representing the possible merge operations) valued with the $loss(C_j C_{j+1})$ as defined above. In each iteration we need to extract the pair of min value from the priority queue, and change the pair (C_{j-1}, C_j) with the pair $(C_{j-1}, (C_j \cup C_{j+1}))$ and the pair (C_j, C_{j+1}) with the pair $((C_j \cup C_{j+1}), C_{j+2})$ together with their values. If we use a binary heap to implement the priority queue each such operation on the priority queue can be done in time $O(g + \log n_i)$.

Therefore, Ratio-Greedy can be implemented to run in $O(n \log n + ng)$ time. Moreover, the impurity of the partition obtained by Ratio-Greedy is no worse than that obtained by $\mathcal{A}^{Dom}$, thus it inherits its approximation guarantees.

7.1 Experimental Evaluation

We tested Ratio-Greedy on the 20NEWS and RCV1 datasets. The former has been previously used to evaluate text classification methods [6, 26, 23, 13]. In particular, we report comparisons with the Divisive Clustering algorithm of Dhillon et al. [13] which (in the terms of our paper) is designed to optimize the difference between the impurity of the classification and the sum of the impurity of the input vectors.

Divisive Clustering is an adaptation of the k-means method that employs KL-divergence rather than Euclidean squared distance. When $k > g$, the initialization of Divisive Clustering resembles $\mathcal{A}^{Dom}$ since it consists of splitting the vectors that are i-dominant among k/g clusters. For $k \leq g$ the initialization is not well specified so that we use $\mathcal{A}^{Dom}$ in this case. In [13] it was shown that Divisive Clustering outperforms the agglomerative clustering methods of [6, 26].

We used the version of the 20 newsgroup data set available in Scikit-learn `https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html`, comprising 18.846 documents evenly divided into 20 disjoint classes. The RCV1-v2 corpus was obtained from `http://www.jmlr.org/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm`. It includes 804,414 documents assigned to 103 different classes (a single document may belong to more than one class).

By using the scikit-learn `CountVectorizer` function, each document is turned into a vector d whose component d_i counts the number of times that word i appears in document d. For this vectorization of the texts, a filter function has been employed to strip out newsgroup-related metadata (headers, footers and quotes), and prune words occurring in less than two documents or in more than 95% of the documents, and english stopwords (for this we used the standard scikit-learn stopword list).

After vectorizing the documents, the following quantities are computed in order to define the input probability vectors for the classification algorithms:

- for each class c, let $p(c) = \frac{|c|}{\sum_c' |c'|}$, where $|c'|$ denotes the number of documents in class c';
- for each word w and each class c let $n(w,c)$ be the number of times the word w appears in documents of class c.
- for each word w and each class c, let $p(w|c) = \frac{1+n(w,c)}{\sum_w (1+n(w,c))}$ represent the probability of a word, conditioned to a given class;
- set $p(w,c) = p(w|c)p(c)$.

To each word a probability vector $\mathbf{w}$ is associated, setting, for each $c = 1, \ldots 20$, $w_c = p(w,c)$. Since $p(w,c) = p(w|c)p(c) = p(c|w)p(w)$, each component w_c represents the probability of class c given the occurrence of word w weighted by the probability of w.

For 20NEWS the input obtained consisted of 51.840 vectors (1 for each word) of dimension 20 (1 for each class). For RCV1 we have 170.946 vector, each of them with 103 components.

The preprocessing was done in Python and the algorithms were developed and run in C on a MacBook Air 2012 with OS X Yosemite (10.10.5), Intel Core i5 1.7 GHz processor and 4 GB of memory RAM 1600 Mhz. The code as well as the datasets are available in `https://github.com/lmurtinho/RatioGreedyClustering/tree/ICML_submission`

The Results. Figure 1 shows the impurities of the partitions obtained for different values of k for both datasets. DC-Init, DC-Iter1 and DC-All correspond, respectively, to different points in the execution of Divisive Clustering: right after its initialization, after its first iteration and at the end. We set a limit of 100 iterations for Divisive Clustering. For both datasets, we observe that Ratio-Greedy obtains partitions clearly better than that of DC-Init. With respect to DC-Iter1, it produces similar results for 20NEWS while for RCV1 it is significantly better when the number of clusters gets larger.

The key advantage of Ratio-Greedy, however, is its execution time. Table 1 shows the times (in seconds) spent by the methods for RCV1 dataset. We can see that execution time of Ratio-Greedy, in contrast with Divisive Clustering, is barely affected by the number of clusters k. When k is large the difference between the methods is quite significant. As an example for $k = 2000$, Ratio-Greedy is approximately 50 and 5000 times faster than DC-Iter1 and DC-All, respectively. For NEWS20 dataset we observed a similar beheviour.

These experiments suggest that Ratio-Greedy is a strong candidate to be used in applications in which the $O(ngk)$ time complexity of DC per iteration may be prohibitive.

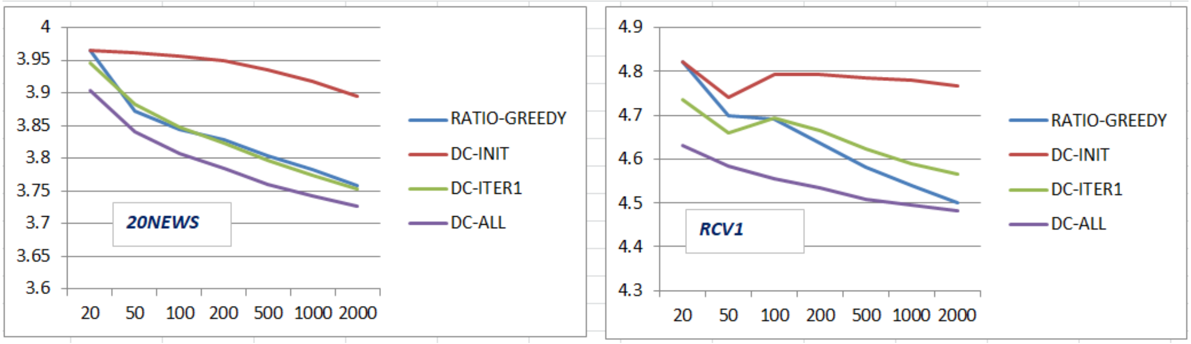

Figure 1: Impurities (vertical axis) of the partitions obtained by Ratio-Greedy and Divisive Clustering for different values of k (horizontal axis).

Table 1: Elapsed time in seconds taken by Divisive Clustering and Ratio-Greedy for RCV1 input.

Clusters	Ratio-Greedy	DC-Init	DC-Iter1	DC-ITER5	DC-All
20	0.6	0.4	3	11	68.6
50	1	0.4	6	25.3	342.1
100	3.1	0.5	10.8	49.1	971.1
200	3.1	0.5	20.3	96.7	1932.4
500	3.3	0.5	48.8	238.7	4823.4
1000	3.5	0.5	96.6	477.2	9612.6
2000	3.5	0.5	191.3	958	19320.2

References

[1] M. R. Ackermann and J. Blömer. Coresets and approximate clustering for bregman divergences. In C. Mathieu, editor, *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 1088–1097. SIAM, 2009.

[2] M. R. Ackermann, J. Blömer, and C. Scholz. Hardness and non-approximability of bregman clustering problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:15, 2011.

[3] M. R. Ackermann, J. Blömer, and C. Sohler. Clustering for metric and non-metric distance measures. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 799–808, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[4] M. R. Ackermann, J. Blömer, and C. Sohler. Clustering for metric and nonmetric distance measures. *ACM Trans. Algorithms*, 6(4):59:1–59:26, 2010.

[5] P. Awasthi, M. Charikar, R. Krishnaswamy, and A. K. Sinop. The hardness of approximation of euclidean k-means. *CoRR*, abs/1502.03316, 2015.

[6] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103. ACM Press, 1998.

[7] L. Breiman, J. J. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[8] D. Burshtein, V. D. Pietra, D. Kanevsky, and A. Nadas. Minimum impurity partitions. *Ann. Statist.*, 1992.

[9] K. Chaudhuri and A. McGregor. Finding metric structure in information theoretic clustering. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 391–402. Omnipress, 2008.

[10] M. Chlebík and J. Chlebíková. Complexity of approximating bounded variants of optimization problems. *Theor. Comput. Sci.*, 354(3):320–338, 2006.

[11] P. A. Chou. Optimal partitioning for classification and regression trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4), 1991.

[12] D. Coppersmith, S. J. Hong, and J. R. M. Hosking. Partitioning nominal attributes in decision trees. *Data Min. Knowl. Discov*, 3(2):197–217, 1999.

[13] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:1265–1287, 2003.

[14] C. Hennig, M. Meila, F. Murtagh, and R. Rocci. *Handbook of Cluster Analysis*. Chapman and Hall/CRC, 2015.

[15] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, Sept. 1999.

[16] S. Jegelka, S. Sra, and A. Banerjee. Approximation algorithms for bregman co-clustering and tensor clustering. *CoRR*, abs/0812.0389, 2008.

[17] A. Kartowsky and I. Tal. Greedy-merge degrading has optimal power-law. *CoRR*, abs/1703.04923, 2017.

[18] B. M. Kurkoski and H. Yagi. Quantization of binary-input discrete memoryless channels. *IEEE Trans. Information Theory*, 60(8):4544–4552, 2014.

[19] E. Laber, M. Molinaro, and F. M. Pereira. Binary partitions with approximate minimum impurity. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2854–2862, Stockholmsmassan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[20] E. S. Laber and L. Murtinho. Minimization of gini impurity: Np-completeness and approximation algorithms via connections with the k-means problem. In *Proceedings of LAGOS*, 2019. To appear.

[21] T. Li, S. Ma, and M. Ogihara. Entropy-based criterion in categorical clustering. In C. E. Brodley, editor, *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004.

[22] M. Lucic, O. Bachem, and A. Krause. Strong coresets for hard and soft bregman clustering with applications to exponential family mixtures. In A. Gretton and C. C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1–9, Cadiz, Spain, 09–11 May 2016. PMLR.

[23] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. 1998.

[24] B. Nazer, O. Ordentlich, and Y. Polyanskiy. Information-distilling quantizers. In *ISIT*, pages 96–100. IEEE, 2017.

[25] U. Pereg and I. Tal. Channel upgradation for non-binary input alphabets and macs. *IEEE Trans. Information Theory*, 63(3):1410–1424, 2017.

[26] N. Slonim and I. N. Tishby. The power of word clusters for text classification. 2001.

[27] N. Slonim and N. Tishby. Agglomerative information bottleneck. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *NIPS*, pages 617–623. The MIT Press, 1999.

[28] I. Tal and A. Vardy. How to construct polar codes. *IEEE Trans. Information Theory*, 59(10):6562–6582, 2013.

[29] A. Vattani. k-means requires exponentially many iterations even in the plane. *Discrete & Computational Geometry*, 45(4):596–616, Jun 2011.

A The proof of Fact 1

Fact 1. *The Gini impurity measure defined by $I_{Gini}(\mathbf{u}) = \|\mathbf{u}\|_1 \sum_{i=1}^{g} \frac{u_i}{\|\mathbf{u}\|_1}(1-\frac{u_i}{\|\mathbf{u}\|_1})$ and the Entropy impurity measure defined by $I_{Ent} = \|\mathbf{u}\|_1 \sum_{i=1}^{g} \frac{u_i}{\|\mathbf{u}\|_1} \log(\frac{\|\mathbf{u}\|_1}{u_i})$ belong to $\mathcal{C}$. For f_{Entr}, a simple inspection shows that (P3) holds at equality.*

Proof. The measure I_{Gini} is obtained using the function $f_{Gini}(x) = x(1-x)$, and I_{Ent} is obtained using the function $f_{Entr}(x) = x \log \frac{1}{x}$. Clearly both functions satisfy property (P1), and it is known they also satisfy (P2) [12]. So it remains to be shown that they satisfy property (P3).

For f_{Gini}, (P3) becomes

$$p(1-p) \leq p(1-q) + p\left(1 - \frac{p}{q}\right) \qquad \forall q \in [p, 1]$$

which after canceling the p's out and rearranging, is equivalent to $p \geq q + \frac{p}{q} - 1$ for all $q \in [p, 1]$, or $p \geq \max_{q \in [p,1]}(q + \frac{p}{q} - 1)$. But the function in the max is convex in q, and hence its maximum is attained at one of the endpoints $q = p$ and $q = 1$; for these endpoints the inequality holds at equality, which then proves the desired property.

For the function $f_{Entr}(x) = -x \log x$ we have that for any $0 < x \leq y < 1$ it holds that

$$-\frac{x}{y} y \log(y) - y\frac{x}{y} \log(\frac{x}{y}) = -x\log(y) - x\log(x) + x\log(y) = -x\log(x),$$

showing that $f_{Entr}(x) = -x \log x$ satisfies (P3) with equality. ∎

B The proof of the Claim in Lemma 8

Claim. Fix $\mathbf{u} \in \mathbb{R}^g$ such that $u_i \geq u_{i+1}$ for each $i = 1, \dots, g-1$. Let $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$ two orthogonal vectors from $\{0,1\}^g \setminus \{\mathbf{0}\}$. Let $i^* = \min\{i \mid \max\{z_i^{(1)}, z_i^{(2)}\} = 1\}$ and $\mathbf{v}^{(1)} = \mathbf{e}_{i^*}$ and $\mathbf{v}^{(2)} = \mathbf{z}^{(1)} + \mathbf{z}^{(2)} - \mathbf{e}_{i^*}$. Then

$$I(\mathbf{u} \circ \mathbf{v}^{(1)}) + I(\mathbf{u} \circ \mathbf{v}^{(2)}) \leq I(\mathbf{u} \circ \mathbf{z}^{(1)}) + I(\mathbf{u} \circ \mathbf{z}^{(1)}).$$

Proof. For the sake of simplifying the notation, let us assume that $i^* = 1$. Since $\mathbf{v}^{(1)} + \mathbf{v}^{(2)} = \mathbf{z}^{(1)} + \mathbf{z}^{(2)}$, and the only significant components are the non-zero components of $\mathbf{z}^{(1)} + \mathbf{z}^{(2)}$, for the analysis, we assume without loss of generality that $\mathbf{z}^{(2)} = \mathbf{1} - \mathbf{z}^{(1)}$. Setting $\mathbf{d} = \mathbf{z}^{(1)}$, we have to prove that

$$I_{Gini}(\mathbf{u} \circ \mathbf{e}_1) + I_{Gini}(\mathbf{u} \circ (\mathbf{1} - \mathbf{e}_1)) \leq I_{Gini}(\mathbf{u} \circ \mathbf{d}) + I_{Gini}(\mathbf{u} \circ (\mathbf{1} - \mathbf{d})),$$

for every $\mathbf{d} \in \{0,1\}^g \setminus \mathbf{0}$.

It follows from the definition of $I_{Gini}(\cdot)$ that

$$I_{Gini}(\mathbf{u}\circ\mathbf{d}) + I_{Gini}(\mathbf{u}\circ(\mathbf{1}-\mathbf{d})) = (\mathbf{u}\cdot\mathbf{d}) \left(\frac{(\mathbf{u}\cdot\mathbf{d})^2 - \sum_{i|d_i=1}(u_i)^2}{(\mathbf{u}\mathbf{d})^2} \right) + (\mathbf{u}(\mathbf{1}-\mathbf{d})) \left(\frac{(\mathbf{u}(\mathbf{1}-\mathbf{d}))^2 - \sum_{i|d_i=0}(u_i)^2}{(\mathbf{u}(\mathbf{1}-\mathbf{d}))^2} \right) =$$

$$\|\mathbf{u}\|_1 - \left(\frac{\sum_{i|d_i=1}(u_i)^2}{\mathbf{u}\cdot\mathbf{d}} \right) - \left(\frac{\sum_{i|d_i=0}(u_i)^2}{\mathbf{u}(\mathbf{1}-\mathbf{d})} \right)$$

Define $g(\mathbf{d})$ as the sum of two last terms of the above expression, that is,

$$g(\mathbf{d}) = \left(\frac{\sum_{i|\mathbf{d}_i=1}(u_i)^2}{\mathbf{u}\cdot\mathbf{d}} \right) + \left(\frac{\sum_{i|\mathbf{d}_i=0}(u_i)^2}{\mathbf{u}(\mathbf{1}-\mathbf{d})} \right)$$

It is enough to prove that $g(\mathbf{e}_1) \geq g(\mathbf{d})$ for an arbitrary $\mathbf{d}$. For that, we assume w.l.o.g. that $d_1 = 1$ due to the symmetry of $g(\mathbf{d})$ with respect to $\mathbf{d}$.

Let

$$\alpha = \frac{\sum_{i>1|d_i=1}(u_i)^2}{\sum_{i>1|d_i=1} u_i} \quad \text{and} \quad \beta = \frac{\sum_{i|d_i=0}(u_i)^2}{\sum_{i|d_i=0} u_i}$$

Thus,

$$g(\mathbf{d}) = \frac{(u_1)^2 + \alpha(\mathbf{u}\cdot\mathbf{d} - u_1)}{u_1 + (\mathbf{u}\cdot\mathbf{d} - u_1)} + \beta$$

Moreover, we can write $g(\mathbf{e}_1)$ as a function of $\mathbf{d}$

$$g(\mathbf{e}_1) = u_1 + \frac{\alpha(\mathbf{u}\cdot\mathbf{d} - u_1) + \beta\mathbf{u}(\mathbf{1} - \mathbf{d})}{(\mathbf{u}\cdot\mathbf{d} - u_1) + \mathbf{u}(\mathbf{1} - \mathbf{d})}$$

The following inequalities will be useful: $\alpha, \beta \leq u_1$ since $u_1 \geq u_i$ for all i, $(\mathbf{u}\cdot\mathbf{d} - u_1) \geq \alpha$ and $\mathbf{u}(\mathbf{1} - \mathbf{d}) \geq \beta$.

We need to prove that

$$g(\mathbf{e}_1) = u_1 + \frac{\alpha(\mathbf{u}\cdot\mathbf{d} - u_1) + \beta(\mathbf{u}(\mathbf{1} - \mathbf{d}))}{(\mathbf{u}\cdot\mathbf{d} - u_1) + \mathbf{u}(\mathbf{1} - \mathbf{d})} \geq \frac{(u_1)^2 + \alpha(\mathbf{u}\cdot\mathbf{d} - u_1)}{u_1 + (\mathbf{u}\cdot\mathbf{d} - u_1)} + \beta = g(\mathbf{d}),$$

or equivalently,

$$\frac{u_1(\mathbf{u}\cdot\mathbf{d} - u_1)}{u_1 + (\mathbf{u}\cdot\mathbf{d} - u_1)} - \frac{\alpha(\mathbf{u}\cdot\mathbf{d} - u_1)}{u_1 + (\mathbf{u}\cdot\mathbf{d} - u_1)} \geq \frac{\beta(\mathbf{u}\cdot\mathbf{d} - u_1)}{\mathbf{u}\cdot(\mathbf{1} - \mathbf{d}) + (\mathbf{u}\cdot\mathbf{d} - u_1)} - \frac{\alpha(\mathbf{u}\cdot\mathbf{d} - u_1)}{\mathbf{u}\cdot(\mathbf{1} - \mathbf{d}) + (\mathbf{u}\cdot\mathbf{d} - u_1)}$$

Simplifying the terms we need to prove

$$(\beta - \alpha)[(\mathbf{u}\cdot\mathbf{d} - u_1) + u_1] \leq (u_1 - \alpha)[(\mathbf{u}\cdot\mathbf{d} - u_1) + \mathbf{u}\cdot(\mathbf{1} - \mathbf{d})]$$

which is equivalent to

$$\beta u_1 - \alpha u_1 \leq (u_1 - \beta)(\mathbf{u}\cdot\mathbf{d} - u_1) + (u_1 - \alpha)\mathbf{u}\cdot(\mathbf{1} - \mathbf{d}), \tag{74}$$

However, because $\alpha, \beta \leq u_1$, $(\mathbf{u}\cdot\mathbf{d} - u_1) \geq \alpha$ and $\mathbf{u}\cdot(\mathbf{1} - \mathbf{d}) \geq \beta$, we have

$$(u_1 - \beta)\alpha + (u_1 - \alpha)\beta \leq (u_1 - \beta)(\mathbf{u}\cdot\mathbf{d} - u_1) + (u_1 - \alpha)\mathbf{u}\cdot(\mathbf{1} - \mathbf{d}).$$

Thus, to establish inequality (74), it is enough to prove that

$$\beta u_1 - \alpha u_1 \leq (u_1 - \beta)\alpha + (u_1 - \alpha)\beta,$$

or, equivalently,

$$\alpha\beta \leq \alpha u_1.$$

The last inequality holds because $u_1 \geq \beta$. □

C The proof of Proposition 3

Proposition 3. *Fix $i \in [g]$ and let B be a set of vector in $\mathbb{R}^g$ such that for each $\mathbf{v} \in B$, it holds that $\|\mathbf{v}\|_\infty = v_i$, i.e., B is i-pure. It holds that*

$$\frac{1}{2} I_2(B) \leq I(B) \leq 2 I_2(B) + 4(\log g) \sum_{\mathbf{w}\in B} I(\mathbf{w}).$$

Proof. Let us assume w.l.o.g. that B is 1-pure. Let $\mathbf{v}$ be the vector corresponding to B, that is, $\mathbf{v} = \sum_{\mathbf{v}' \in B} \mathbf{v}'$. Moreover, let

$$\mathbf{u} = \sum_{\mathbf{v}' \in B} \chi^{2C}(\mathbf{v}')$$

,

$$\mathbf{u}^k = \sum_{\mathbf{v}' \in B : \|\mathbf{v}'\|_\infty < \|\mathbf{v}\|_1/2} \chi^{2C}(\mathbf{v}')$$

and

$$\mathbf{u}^H = \sum_{\mathbf{v}' \in B : \|\mathbf{v}\|_\infty \geq \|\mathbf{v}\|_1/2} \chi^{2C}(\mathbf{v}')$$

Note that $\mathbf{u}^k$ corresponds to the set of vectors for which the dominant component is affected by transformation χ^{2C}. It shall be clear that $\|\mathbf{v}\|_1 = \|\mathbf{u}\|_1$ and

$$\|\mathbf{v}\|_\infty \leq \|\mathbf{u}^k\|_\infty + \|\mathbf{u}^H\|_\infty = \frac{\|\mathbf{u}^k\|_1}{2} + \|\mathbf{u}^H\|_\infty = \|\mathbf{u}\|_\infty$$

From Lemma 9 and Corollary 1 we have that

$$(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \max \left\{ 1, \log \left(\frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty} \right) \right\} \leq I(\mathbf{v}) \leq 2(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \left(\frac{2g\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty} \right) \tag{75}$$

Let $\alpha = \|\mathbf{u}\|_1 - \|\mathbf{u}^H\|_\infty - \frac{\|\mathbf{u}^k\|_1}{2}$. Then, we have

$$I(\mathbf{u}) = \alpha \log \left(\frac{\|\mathbf{u}\|_1}{\alpha} \right) + (\|\mathbf{u}\|_1 - \alpha) \log \left(\frac{\|\mathbf{u}\|_1}{\|\mathbf{u}\|_1 - \alpha} \right) ‘ \tag{76}$$

Since $\|\mathbf{u}^H\|_\infty + \frac{\|\mathbf{u}^k\|_1}{2} \geq \frac{\|\mathbf{u}\|_1}{2}$ then $\alpha \leq \frac{\|\mathbf{u}\|_1}{2}$, from Proposition 1 we have $\frac{\|\mathbf{u}\|_1 - \alpha}{\|\mathbf{u}\|_1} \log \frac{\|\mathbf{u}\|_1}{\|\mathbf{u}\|_1 - \alpha} \leq \frac{\alpha}{\|\mathbf{u}\|_1} \log \frac{\|\mathbf{u}\|_1}{\alpha}$. This, together with (76) implies that

$$\alpha \log \frac{\|\mathbf{u}\|_1}{\alpha} \leq I(\mathbf{u}) \leq 2\alpha \log \frac{\|\mathbf{u}\|_1}{\alpha}. \tag{77}$$

Now we note that $\|\mathbf{u}^H\|_\infty + \frac{\|\mathbf{u}^k\|_1}{2} > \|\mathbf{v}\|_\infty$, hence

$$\alpha = \|\mathbf{u}\|_1 - \|\mathbf{u}^H\|_\infty - \frac{\|\mathbf{u}^k\|_1}{2} \leq \|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty. \tag{78}$$

We first focus on the proof of the left bound $\frac{1}{2} I_2(B) \leq I(B)$. We split the analysis into two cases

Case 1. $\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty \leq \frac{\|\mathbf{v}\|_1}{e}$. Then

$$I(\mathbf{u}) \leq 2\alpha \log \frac{\|\mathbf{v}\|_1}{\alpha} \leq 2(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log \frac{\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty} \leq 2I(\mathbf{v}) \tag{79}$$

where the first inequality is from (77), the second inequality is from Proposition 2 and the last inequality is from (75) (using the hypothesis at the basis of this case).

Case 2. $\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty > \frac{\|\mathbf{v}\|_1}{e}$. Then,

$$I(\mathbf{v}) \geq (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log e \geq \|\mathbf{v}\|_1 = \|\mathbf{u}\|_1 \geq I(\mathbf{u}) \tag{80}$$

where the first inequality is from (75) and the last inequality is because by definition of $I = I_{Ent}$, for a 2 dimensional vector $\mathbf{u}$ we have $I(\mathbf{u}) \leq \|\mathbf{u}\|_1$.

From (79) and (79) it immediately follows that $\frac{1}{2}I_2(B) = \frac{1}{2}I(\mathbf{u}) \leq I(\mathbf{v}) = I(B)$.

We now focus on the right inequality and show that $I(\mathbf{v}) \leq 2I(\mathbf{u}) + 4(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log g$, from which also the last inequality in the statement of the proposition immediately follows.

First, we observe that

$$\begin{aligned} 2\left(\|\mathbf{u}\|_1 - \|\mathbf{u}^H\|_\infty - \frac{\|\mathbf{u}^k\|_1}{2}\right) &= \|\mathbf{u}^k\|_1 + 2(\|\mathbf{u}^H\|_1 - \|\mathbf{u}^H\|_\infty) \geq (\|\mathbf{u}^k\|_1 + (\|\mathbf{u}^H\|_1 - \|\mathbf{u}^H\|_\infty)) && (81) \\ &= \|\mathbf{u}\|_1 - \|\mathbf{u}^H\|_\infty \geq \|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty && (82) \end{aligned}$$

Therefore, usign (81)(82) we have $2\left(\|\mathbf{u}\|_1 - \|\mathbf{u}^H\|_\infty - \frac{\|\mathbf{u}^k\|_1}{2}\right) \leq \frac{2g\|\mathbf{u}\|_1}{e}$, hence from the right inequality of (75) we get

$$\begin{aligned} I(\mathbf{v}) &\leq 2\left(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty\right) \log\left(\frac{2g\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty}\right) && (83) \\ &\leq 4\left(\|\mathbf{u}\|_1 - \|\mathbf{u}^H\|_\infty - \frac{\|\mathbf{u}^k\|_1}{2}\right) \log \frac{2g\|\mathbf{v}\|_1}{2\left(\|\mathbf{u}\|_1 - \|\mathbf{u}^H\|_\infty - \frac{\|\mathbf{u}^k\|_1}{2}\right)} && (84) \\ &= 4\left(\|\mathbf{u}\|_1 - \|\mathbf{u}^H\|_\infty - \frac{\|\mathbf{u}^k\|_1}{2}\right)\left(\log \frac{\|\mathbf{v}\|_1}{\left(\|\mathbf{u}\|_1 - \|\mathbf{u}^H\|_\infty - \frac{\|\mathbf{u}^k\|_1}{2}\right)} + \log g\right) && (85) \\ &\leq 2I(\mathbf{u}) + 4\left(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty\right) \log g && (86) \end{aligned}$$

where the last inequality follows from the left hand side of (76) together with the definition of α (for the first term) and from $\|\mathbf{u}^H\|_\infty - \frac{\|\mathbf{u}^k\|_1}{2} \geq \|\mathbf{v}\|_\infty$ and $\|\mathbf{v}\|_1 = \|\mathbf{u}\|_1$ (for the second term).

Since B is i-pure, we have that $\|v\|_1 = \sum_{\mathbf{w}\in B} \|\mathbf{w}\|_1$ and $\|v\|_\infty = \sum_{\mathbf{w}\in B} \|\mathbf{w}\|_\infty$. Then, we have

$$I(\mathbf{v}) \leq 2I(\mathbf{u}) + 4(\log g) \sum_{\mathbf{w}\in B} (\|\mathbf{w}\|_1 - \|\mathbf{w}\|_\infty) \leq 2I(\mathbf{u}) + 4(\log g) \sum_{\mathbf{w}\in B} I(\mathbf{w})$$

where the last inequality follows by Corollary 1.

We have then shown the right inequalities of the statement. The proof of the Proposition is complete.

□

D Proof of Lemma 14

Proof. Recall that $s_{i,mix}$ denotes the total sum of the components of the i-dominant vectors from bucket B_{mix} and that we assumed $dom(B_{mix}) = 1$. We let $c_{i,mix} = s_{i,mix} - \sum_{\mathbf{v}\in V_{i,mix}} v_1$, i.e., the total sum of all but the first component of the i-dominant vectors in B_{mix}. Moreover, let

$s_{mix} = \sum_{i=1}^{g} s_{i,mix}$ and $c_{mix} = \sum_{i=1}^{g} c_{i,mix}$. Furthermore, let $c_{i,p} = s_{i,p} - \sum_{\mathbf{v} \in B_i} v_i$, i.e., the total sum of the non-i components of vectors in B_i.

It follows from Corollary 1 that

$$I(B_{mix}) \geq c_{mix} \max\left\{1, \log\left(\frac{s_{mix}}{c_{mix}}\right)\right\} \tag{87}$$

Note that for $i > 1$, we have $c_{i,mix} \geq (s_{i,mix})/2$, for otherwise i would not be the dominant component in $V_{i,mix}$. Thus, we also have that

$$I(B_{mix}) \geq c_{mix} \geq c_{1,mix} + \sum_{i=2}^{g} s_{i,mix}/2 \tag{88}$$

Moreover, if $c_{mix} < s_{mix}/e$, from (87) we have that

$$I(B_{mix}) \geq c_{mix} \log\left(\frac{s_{mix}}{c_{mix}}\right) \geq \left(c_{1,mix} + \sum_{i=2}^{g} s_{i,mix}/2\right) \log\left(\frac{s_{mix}}{c_{1,mix} + \sum_{i=2}^{g} s_{i,mix}/2}\right), \tag{89}$$

where the last inequality follows from (88) and Proposition 2.

From Corollary 1 we have that

$$I(B_i) \geq c_{i,p} \max\left\{1, \log\left(\frac{s_{i,p}}{c_{i,p}}\right)\right\} \tag{90}$$

for every i-pure bucket B_i

Now we derive upper bounds on $B'_{mix}, B'_1, \ldots, B'_g$ and compare them with the lower bounds given by the previous equations.

Bound on the mixed bucket B'_{mix}.

Let $s^k_{i,mix} = \|V_{i,mix} \cap B'_{mix}\|_1$, this is the total sum of the components of the i-dominant vectors in $B_{mix} \cap B'_{mix}$.

Moreover, let

$$c^k_{i,mix} = s^k_{i,mix} - \sum_{\mathbf{v} \in V_{i,mix} \cap B'_{mix}} v_1,$$

i.e., the total sum of all but the first components in the i-dominant vectors in $B_{mix} \cap B'_{mix}$.

Recall that Y_i is the set of i-dominant vectors moved from B_i to B_{mix} in order to obtain partition $\mathcal{P}'$. Let $s^k_{i,p} = \|Y_i\|_1$ and let

$$t^k_{i,p} = s^k_{i,p} - \sum_{\mathbf{v} \in Y_i} v_1,$$

i.e., the total sum of all but the first components of vectors in Y_i.

In these notations, the superscript k is used to remind the reader that these quantities refer to vectors with 'low' ratio.

From Corollary 1 (with $i = 1$) we have that

$$I(B'_{mix}) \leq 2\left(\sum_{i=1}^{g} c^k_{i,mix} + t^k_{i,p}\right) \log\left(\frac{2g\left(\sum_{i=1}^{g} s^k_{i,mix} + s^k_{i,p}\right)}{\sum_{i=1}^{g} c^k_{i,mix} + t^k_{i,p}}\right) \tag{91}$$

Moreover, we have

$$\sum_{i=1}^{g}(c^k_{i,mix}+t^k_{i,p}) \le c^k_{1,mix}+t^k_{1,p}+\sum_{i=2}^{g}(s^k_{i,mix}+s^k_{i,p}) \tag{92}$$

Thus, we have that

$$\begin{aligned}
I(B'_{mix}) &\le 2\left(c^k_{1,mix}+t^k_{1,p}+\sum_{i=2}^{g}(s^k_{i,mix}+s^k_{i,p})\right)\log\left(\frac{2g\left(\sum_{i=1}^{g}s^k_{i,mix}+s^k_{i,p}\right)}{c^k_{1,mix}+t^k_{1,p}+\sum_{i=2}^{g}(s^k_{i,mix}+s^k_{i,p})}\right)\\
&\le 2\left(c^k_{1,mix}+\sum_{i=2}^{g}(s^k_{i,mix}+s^k_{i,p})\right)\log\left(\frac{4gs_{mix}}{c^k_{1,mix}+\sum_{i=2}^{g}(s^k_{i,mix}+s^k_{i,p})}\right)\\
&\quad +2t^k_{1,p}\log\left(\frac{2g\sum_{i=1}^{g}(s^k_{i,mix}+s^k_{i,p})}{t^k_{1,p}+\sum_{i=2}^{g}(s^k_{i,mix}+s^k_{i,p})}\right)\\
&\le 2\left(c^k_{1,mix}+2\sum_{i=2}^{g}s_{i,mix}\right)\log\left(\frac{4gs_{mix}}{c^k_{1,mix}+2\sum_{i=2}^{g}s_{i,mix}}\right) &(93)\\
&\quad +2t^k_{1,p}\log\left(\frac{2g\sum_{i=1}^{g}(s^k_{i,mix}+s^k_{i,p})}{t^k_{1,p}+\sum_{i=2}^{g}(s^k_{i,mix}+s^k_{i,p})}\right), &(94)
\end{aligned}$$

where the first inequality follows from inequality (91), Proposition 2 and inequality (92); the second inequality follows from $s^k_{i,p} \le s_{i,mix}$ and the third inequality from $s^k_{i,p} \le s_{i,mix}$ together with Proposition 2.

We prove that the expression in (93)-(94) is at most an $O(\log g)$ factor of $I(B_{mix})+I(B_1)$. First, we consider the term in (93) that we denote by α.

If $c_{mix} \ge s_{mix}/e$ then

$$\begin{aligned}
\alpha &= \left(c^k_{1,mix}+2\sum_{i=2}^{g}s_{i,mix}\right)\log\left(\frac{2gs_{mix}}{c^k_{1,mix}+2\sum_{i=2}^{g}s_{i,mix}}\right)\\
&\le \left(2c_{1,mix}+2\sum_{i=2}^{g}s_{i,mix}\right)\log\left(\frac{2gs_{mix}}{2c_{1,mix}+2\sum_{i=2}^{g}s_{i,mix}}\right)\\
&\le \left(2c_{1,mix}+2\sum_{i=2}^{g}s_{i,mix}\right)\log\left(\frac{2gs_{mix}}{2c_{1,mix}+2\sum_{i=2}^{g}c_{i,mix}}\right)\\
&= \left(2c_{1,mix}+2\sum_{i=2}^{g}s_{i,mix}\right)\log\left(\frac{2gs_{mix}}{2c_{mix}}\right)\\
&\le \left(2c_{1,mix}+2\sum_{i=2}^{g}s_{i,mix}\right)\log(ge) \le 4\left(c_{1,mix}+\sum_{i=2}^{g}\frac{s_{i,mix}}{2}\right)\log(ge) &(95)
\end{aligned}$$

which is at a $O(\log g)$ factor from the lower bound on $I(B_{mix})$ given by inequality (88).

On the other hand, if $c_{mix} < s_{mix}/e$, then the first term of (94) is at $O(\log g)$ factor from lower bound given by inequality (89), in fact we have

$$
\begin{aligned}
\alpha &= \left(c^k_{1,mix} + 2\sum_{i=2}^{g} s_{i,mix}\right)\log\left(\frac{2gs_{mix}}{c^k_{1,mix} + 2\sum_{i=2}^{g} s_{i,mix}}\right)\\
&\le \left(c_{1,mix} + 2\sum_{i=2}^{g} s_{i,mix}\right)\log\left(\frac{2gs_{mix}}{c_{1,mix} + 2\sum_{i=2}^{g} s_{i,mix}}\right)\\
&\le 4\left(c_{1,mix} + \sum_{i=2}^{g} \frac{s_{i,mix}}{2}\right)\log\left(\frac{2gs_{mix}}{c_{1,mix} + \sum_{i=2}^{g} \frac{s_{i,mix}}{2}}\right),
\end{aligned}
$$

where the first inequality follows from Proposition 2.

Now, we turn to the second term of (94), which we will denote here by β. We have that

$$
\begin{aligned}
\beta &= t^k_{1,p}\log\left(\frac{2g(s^k_{1,mix} + s^k_{1,p}) + 2g(\sum_{i=2}^{g} s^k_{i,mix} + s^k_{i,p})}{t^k_{1,p} + \sum_{i=2}^{g}(s^k_{i,mix} + s^k_{i,p})}\right)\\
&\le t^k_{1,p}\log\left(\max\left\{\frac{2g(s^k_{1,mix} + s^k_{1,p})}{t^k_{1,p}}, 2g\right\}\right)\\
&\le t^k_{1,p}\log\left(\max\left\{\frac{4g\cdot s_{1,p}}{t^k_{1,p}}, 2g\right\}\right)\\
&\le c_{1,p}\log\left(\max\left\{\frac{4g\cdot s_{1,p}}{c_{1,p}}, 2g\right\}\right),
\end{aligned}
$$

where the second inequality holds because $s^k_{1,mix} \le s_{1,p}$. Moreover, since $t^k_{1,p} \le c_{1,p}$ the last inequality holds due to Proposition 2. It is now easy to see that the quantity in the righthand side of the last inequality is at a $O(\log g)$ factor from the lower bound on the impurity of B_1 given by inequality 90.

We have completed the proof that $I(B'_{mix}) = O(\log g)(I(B_{mix}) + I(B_1))$ as desired.

Bound on i-pure buckets. Recall that X_i is the set of vectors moved from B_{mix} to B_i in order to obtain partition $\mathcal{P}'$. Let $s^H_{i,mix}$ be the total sum of the components of all vectors in X_i, i.e., $s^H_{i,mix} = \|\sum_{\mathbf{v}\in X_i}\mathbf{v}\|_1$. Let $d^H_{i,mix}$ be the total sum of all but the ith components over all vectors in X_i, i.e., $d^H_{i,mix} = s^H_{i,mix} - \sum_{\mathbf{v}\in X_i} v_i$. In addition, let $s^H_{i,p}$ be the total sum of the components of the vectors in the set $B_i \setminus Y_i$, i.e., $s^H_{i,p} = \|\sum_{\mathbf{v}\in B_i\setminus Y_i}\mathbf{v}\|_1$. Finally, let $c^H_{i,p}$ be the total sum of all but the ith component over all the vectors in $B_i \setminus Y_i$ that is $c^H_{i,p} = s^H_{i,p} - \sum_{\mathbf{v}\in B_i\setminus Y_i} v_i$ and let $c^k_{i,p}$ be the total sum of all but the ith component over all the vectors in Y_i that is $c^k_{i,p} = s^H_{i,p} - \|Y_i\|_\infty$.

Case 1.) $s_{i,p} \ge s_{i,mix}$. From Lemma 9 we have that

$$
I(B'_i) \le 2(c^H_{i,p} + d^H_{i,mix})\log\left(\frac{g\cdot(s^H_{i,mix} + s^H_{i,p})}{c^H_{i,p} + d^H_{i,mix}}\right) \le 2(c^H_{i,p} + d^H_{i,mix})\log\left(\frac{2g\cdot s_{i,p}}{c^H_{i,p} + d^H_{i,mix}}\right) \tag{96}
$$

Let $\mathbf{v}$ be the first vector of bucket B_i that is not moved to B_{mix} and let $s = \|\mathbf{v}\|_1$. In particular, $\mathbf{v}$ is the vector with the smallest ratio in B_i among those that are not moved to B_{mix}. Let $c = s - v_i$.

Recall definition of r_i in the construction of $\mathcal{P}'$. We have that

$$\frac{s^H_{i,mix}}{d^H_{i,mix}} > r_i \geq \frac{s^k_{i,p} + s}{c^k_{i,p} + c}. \tag{97}$$

and

$$s^H_{i,mix} \leq s_{i,mix} \leq s^k_{i,p} + s, \tag{98}$$

where the second inequality follows by the definition of X_i and Y_i under the standing assumption $s_{i,p} \geq s_{i,mix}$.

Thus, from (97) and (98) we conclude that $d^H_{i,mix} \leq c^k_{i,p} + c \leq c_{i,p}$, hence, $c^H_{i,p} + d^H_{i,mix} \leq 2c_{i,p}$.

Therefore, from (96) and Proposition 2 we have

$$I(B'_i) \leq 2(c^H_{i,p} + d^H_{i,mix}) \log\left(\frac{2g \cdot s_{i,p}}{c^H_{i,p} + d^H_{i,mix}}\right) \leq 4c_{i,p} \log\left(\frac{g \cdot s_{i,p}}{c_{i,p}}\right). \tag{99}$$

Case 2.) $s_{i,p} < s_{i,mix}$.

subcase 2.1) $i = 1$. From Lemma 9, we have

$$I(B'_1) \leq 2(c^H_{1,p} + c^H_{1,mix}) \log\left(\frac{g(s^H_{1,p} + s^H_{1,mix})}{c^H_{1,p} + c^H_{1,mix}}\right) \leq 2(c^H_{1,p} + c^H_{1,mix}) \log\left(\frac{2g \cdot s_{mix}}{c^H_{1,p} + c^H_{1,mix}}\right)$$

Let $\mathbf{v}$ be the first vector of B_{mix} that is moved to B_1. Let $s = \|\mathbf{v}\|_1$ and let $c = s - v_1$. By construction we have that $s^H_{1,p} \leq s_{1,p} \leq s^k_{1,mix} + s$.

Moreover,

$$\frac{s^H_{1,p}}{c^H_{1,p}} \geq r_1 \geq \frac{s^k_{1,mix} + s}{c^k_{1,mix} + c},$$

hence $c^H_{1,p} \leq c^k_{1,mix} + c$.

Therefore, $c^H_{1,p} + c^H_{1,mix} \leq 2c_{1,mix}$. Thus, by the above inequality on $I(B'_1)$ and Proposition 2 we have

$$I(B'_1) \leq 4c_{1,mix} \log\left(\frac{2g \cdot s_{mix}}{c_{1,mix}}\right) \leq 4c_{mix} \log\left(\frac{2g \cdot s_{mix}}{c_{mix}}\right). \tag{100}$$

subcase 2.2 $i > 1$.

In this case the bucket B'_i is exactly the set X_i. Thus, it follows from the subadditivity of I that

$$I(B'_i) = I(X_i) \leq I(V_{i,mix}) \tag{101}$$

Thus, by aggregating the upper bounds given by Equations (99), (100) and (101), we get that

$$\sum_{i=1}^{g} I(B'_i) \leq \sum_{i=1}^{g} 4c_{i,p} \log\left(\frac{g \cdot s_{i,p}}{c_{i,p}}\right) + 4c_{mix} \log\left(\frac{2g \cdot s_{mix}}{c_{mix}}\right) + \sum_{i=2}^{g} I(V_{i,mix})$$

The first term is at most $O(\log g) \sum_{i=1}^{g} I(B_i)$ due to the lower bound in (90). The second term is $O(\log g) I(B_{mix})$ due to the lower bounds in (88) and (89). Finally, the last term is at most $I(B_{mix})$ due to the subadditivity of I.

$\square$