New Results on Information Theoretic Clustering

Ferdinando Cicalese [1] **Eduardo Laber** [2] **Lucas Murtinho** [2]

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 (Hennig et al., 2015; Jain et al., 1999). 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., (Dhillon et al., 2003) 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 (Breiman et al., 1984):

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

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

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.,

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

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

[*] Equal contribution [1] Department of Computer Science, University of Verona, Verona, Italy [2] Departamento de Informática, PUC-RIO, Rio de Janeiro, Brazil. Correspondence to: Ferdinando Cicalese <ferdinando.cicalese@univr.it>, Eduardo Laber <eduardo.laber1@gmail.com>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

construction of random forest/decision trees (Breiman et al., 1984; Burshtein et al., 1992; Chou, 1991; Coppersmith et al., 1999; Laber et al., 2018). (ii) clustering of words based on their distribution over a text collection for improving classification tasks (Baker & McCallum, 1998; Dhillon et al., 2003) and (iii) quantization of memoryless channels/design of polar codes (Tal & Vardy, 2013; Kurkoski & Yagi, 2014; Kartowsky & Tal, 2017; Pereg & Tal, 2017; Nazer et al., 2017). 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 (Chou, 1991).

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 (Awasthi et al., 2015) 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 (Awasthi et al., 2015; Laber & Murtinho, 2019. To appear).

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 (Chaudhuri & McGregor, 2008; Ackermann et al., 2011).

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 (Dhillon et al., 2003), 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 (Laber et al., 2018) 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 (Laber et al., 2018), 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 $g = 2$ (Kurkoski & Yagi, 2014);(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 (Kurkoski & Yagi, 2014) showed that PMWIP$_{Ent}$ can be solved in polynomial time when $g = 2$. The correctness of this algorithm relies on a theorem, proved in (Breiman et al., 1984), which is generalized for $g > 2$ and k groups in (Chou, 1991; Burshtein et al., 1992; Coppersmith et al., 1999). 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 $k = 2$ (Laber et al., 2018). 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 (Tal & Vardy, 2013; Kurkoski & Yagi, 2014; Kartowsky & Tal, 2017; Pereg & Tal, 2017; Nazer et al., 2017) 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$_{KL}$ (Chaudhuri & McGregor, 2008), 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 (Chaudhuri & McGregor, 2008) 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 $[\lambda, v]$ (Ackermann et al., 2008; Ackermann & Blömer, 2009; Ackermann et al., 2010; Lucic et al., 2016). The algorithm from (Ackermann et al., 2008; 2010) runs in $O(n2^{O(mk/\epsilon \log(mk/\epsilon))})$ time, where m is a constant that depends on ϵ and λ. In (Ackermann & Blömer, 2009) 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 (Lucic et al., 2016). 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. (Jegelka et al., 2008) show that Lloyds k-means algorithm—which also has an exponential time worst case complexity (Vattani, 2011)—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 (Chaudhuri & McGregor, 2008) 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 (Chaudhuri & McGregor, 2008) 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 (Chaudhuri & McGregor, 2008) depends on the logarithm of the number of input vectors.

In terms of computational complexity, Chaudhuri and McGregor (Chaudhuri & McGregor, 2008) proved that a variant of MTC$_{KL}$ where the centroids must be chosen from the vectors in V is NP-Complete. Ackermann et. al. (Ackermann et al., 2011) 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 (Baker & McCallum, 1998; Coppersmith et al., 1999; Slonim & Tishby, 1999; Dhillon et al., 2003; Li et al., 2004; Lucic et al., 2016). A variant of Loyds k-means that uses Kullback-Leibler divergence rather than squared Euclidean distance was proposed independently in (Chou, 1991; Dhillon et al., 2003). Experiments from (Dhillon et al., 2003) suggest that this method, denoted by them as DIVISIVE CLUSTERING, is superior to those proposed in (Baker & McCallum, 1998; Slonim & Tishby, 1999). 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, I, k), 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 impurity measure.

We assume that for each component $i = 1, \dots, g$ 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 $\mathcal{P} = (V^{(1)}, \dots, V^{(k)})$ of the set V is then $I(\mathcal{P}) = \sum_{i=1}^{k} I(V^{(i)})$. We use $\mathsf{opt}(V, I, k)$ to denote the minimum possible impurity for a 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 $\mathsf{opt}(V)$ (instead of $\mathsf{opt}(V, I, k)$). We say that a partition $(V^{(1)}, \dots, V^{(k)})$ is optimal for input (V, I, k) iff $\sum_{i=1}^{k} I(V^{(i)}) = \mathsf{opt}(V, I, k)$.

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 $\mathbf{u} \circ \mathbf{v} = (u_1 v_1, \ldots, u_g v_g)$ 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 impurity measures we will focus on, namely Gini and Entropy, are special cases of the class of *frequency-weighted impurity measures based on concave functions* (Coppersmith et al., 1999). A fundamental property of such impurity measures is that they are *superadditive* as shown in (Coppersmith et al., 1999). Moreover, Gini and Entropy are members of a certain class of impurity measures $\mathcal{C}$ defined in (Laber et al., 2018). Impurities from this class satisfy a special *subsystem property* which will be used in our analysis to relate the impurity of partitions for instances of dimension g with that for instances of dimension k when $g > k$.

Lemma 2.1 (Subsystem Property, (Laber et al., 2018))**.** *Let I be an impurity measure in $\mathcal{C}$. Then, for every $\mathbf{u} \in \mathbb{R}^g_+$ and pairwise orthogonal vectors $\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(k)} \in \{0,1\}^g$, such that $\sum_{i=1}^k \mathbf{d}^{(i)} = \mathbf{1}$, we have*

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

Moreover, for $I = I_{Ent}$ we have that (2) holds with equality.

All the proofs can be found in the supplementary material.

3. Handling High-dimensional Vectors

In this section we present an approach to address instances (V, I, k) with $I \in \{I_{Gini}, I_{Ent}\}$ 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 (Laber et al., 2018) 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 $D \in \mathcal{D}$ 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 $(proj_D(V), I, k)$ and the corresponding approximation attained by $(V^{(1)}, \ldots, V^{(k)})$ for instance (V, I, k). In the rest of the section, we assume I and k fixed and we omit to specify them.

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 and the superadditivity of I we have the following upper bound on the impurity of the partition returned by $\mathcal{A}$ on instance (V, I, k).

$$I(\mathcal{A}) \leq \sum_{i=1}^{k} I\left((\mathbf{u}^{(i)} \cdot \mathbf{d}^{(1)}, \ldots, \mathbf{u}^{(i)} \cdot \mathbf{d}^{(k)})\right) + \sum_{\mathbf{d} \in D} I(\mathbf{u} \circ \mathbf{d}). \quad (3)$$

Let D be an arbitrary sequence in $\mathcal{D}$. We have the following lower bounds on $\mathsf{opt}(V)$:

$$\mathsf{opt}(V) \geq \max\{\mathsf{opt}(proj_D(V)), \min_{D' \in \mathcal{D}} \sum_{\mathbf{d}' \in D'} I(\mathbf{u} \circ \mathbf{d}')\} \quad (4)$$

The proof of the first bound in max is obtained using the fact that the impurities I_{Ent} and I_{Gini} are sums of subadditive functions. To prove the second bound in max we first use the fact that $\mathsf{opt}(V)$ can be lower bounded by $\mathsf{opt}(V')$ where V' is an instance with nk vectors obtained by creating k vectors $v_1 \mathbf{e}_1, \ldots v_k \mathbf{e}_k$ for each $\mathbf{v} \in V$. Then, we use the *hyperplane separation* theorem proved in (Burshtein et al., 1992; Coppersmith et al., 1999) to show that $\mathsf{opt}(V')$ corresponds to a partition of the coordinates of $\mathbf{u}$. The result follows because the second argument of max in lower bound (4) considers all partitions of coordinates of $\mathbf{u}$.

Putting together (3) and (4) we have

Lemma 3.1. *Fix an instance (V, I, k) with $g > k$ and also a sequence $D \in \mathcal{D}$. Let $\mathcal{A}$ be an algorithm that outputs partition $\mathcal{P} = (V^{(1)}, \ldots, V^{(k)})$ for instance (V, I, k) and let $proj_D(\mathcal{P}) = (proj_D(V^{(1)}), \ldots, proj_D(V^{(k)})$ be the partition corresponding to $\mathcal{P}$ for the set $proj_D(V)$. It holds that*

$$\frac{I(\mathcal{A})}{\mathsf{opt}(V)} \leq \frac{\sum_{i=1}^{k} I\left((\mathbf{u}^{(i)} \cdot \mathbf{d}^{(1)}, \ldots, \mathbf{u}^{(i)} \cdot \mathbf{d}^{(k)})\right)}{\mathsf{opt}(proj_D(V))} + \frac{\sum_{\mathbf{d} \in D} I(\mathbf{u} \circ \mathbf{d})}{\min_{D \in \mathcal{D}} \sum_{\mathbf{d}' \in D} I(\mathbf{u} \circ \mathbf{d}')} \quad (5)$$

Note that the first ratio in the last expression is the approximation attained by the partition

$proj_D(V^{(1)}), \dots, proj_D(V^{(k)})$ on the instance $proj_D(V)$. Thus, the above inequality says that we can obtain a good approximation for instance (V) of PMWIP by properly choosing a set D of k orthogonal directions in $\{0,1\}^g$, and also—given the choice of D—a good approximation for the instance $(proj_D(V))$ of PMWIP.

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 $u_i \leq u_{i-1}$, for $i = 2, \dots, g$.

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

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

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 $D = \{\mathbf{d}^{(1)}, \dots, \mathbf{d}^{(k)}\} \in \mathcal{D}$ where $\mathbf{d}^{(i)} = \mathbf{e}_i$ for $i = 1, \dots, k-1$ 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 (5) as

$$\frac{I(\mathcal{A}^{Dom}(V))}{\mathsf{opt}(V, I, k)} \leq \frac{I(\mathcal{A}^{Dom}(proj_D(V)))}{\mathsf{opt}(proj_D(V), I, k)} + \frac{\sum_{\mathbf{d}\in D} I(\mathbf{u}\circ\mathbf{d})}{\min_{D\in\mathcal{D}} \sum_{\mathbf{d}'\in D} I(\mathbf{u}\circ\mathbf{d}')} \qquad (6)$$

We can prove that the first term of the righthand side of (6) is upper bounded by $O(\log(\sum_{\mathbf{v}\in V} \|\mathbf{v}\|_1))$. In fact, by using the superaditivity of I we just need to bound, for each group $V^{(i)}$ in the partition obtained by $\mathcal{A}^{Dom}$, the ratio $I_{Ent}(\sum_{\mathbf{v}\in V^{(i)}} \mathbf{v}) / \sum_{\mathbf{v}\in(i)} I_{Ent}(\mathbf{v})$. This can be achieved by using the following bounds on $I(\mathbf{v})$ that depend on the largest component of $\mathbf{v}$

$$(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log\left(\frac{\|\mathbf{v}\|_1}{\min\{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty, \|\mathbf{v}\|_\infty\}}\right) \leq$$
$$I_{Ent}(\mathbf{v}) \leq 2(\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty) \log\left(\frac{g\|\mathbf{v}\|_1}{\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty}\right). \qquad (7)$$

We can also prove that the bound on the second ratio in (6) is $O(\log k)$. The proof makes use of the fact that the sequence $D' \in \mathcal{D}$ that minimizes the denominator in the rightmost ratio of (6) is the one that induces the most balanced partition of the coordinates of $\mathbf{u}$. Therefore, we have

Theorem 4.1. *$\mathcal{A}^{Dom}$ is an $O(\log(\sum_{\mathbf{v}\in V} \|\mathbf{v}\|_1))$-approximation algorithm for* PMWIP$_{Ent}$.

Remark 4.1. *Let s be a large integer. The instance $\{(s,0), (2,1), (0,1)\}$ and $k = 2$ shows that the above 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.

In the full paper we also show that $\mathcal{A}^{Dom}$ guarantees $O(\log n + \log g)$-approximation for the restriction of PMWIP$_{Ent}$ where all vectors in V have the same ℓ_1-norm. Moreover, by reasoning along the same lines we obtain

Theorem 4.2. *Algorithm $\mathcal{A}^{Dom}$ is a linear time 3-approximation for general instances (V, I_{Gini}, k) and a 2-approximation algorithm for restricted instances where $g \leq k$.*

5. Better Approximation for PMWIP$_{Ent}$

In this section we only focus on the entropy impurity measure. We will use I for I_{Ent} and PMWIP for PMWIP$_{Ent}$.

Under the assumption $g \leq k$, we will show the existence of an $O(\log^2 g)$-approximation polynomial time algorithm for PMWIP. Note that due to the approach of Section 3 (see, in particular equation (5)), this implies an $O(\log^2(\min\{g,k\}))$-approximation algorithm for any g and k. We will assume here that vectors in V have non-negative integer coordinates.

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 $\|\sum_{\mathbf{v}\in B} \mathbf{v}\|_\infty$. 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., $\|B\|_\infty = \sum_{\mathbf{v}\in B} v_i$. 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 inequality (7), we define the *ratio of a vector* $\mathbf{v}$ as $ratio(\mathbf{v}) = \|\mathbf{v}\|_1 / (\|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)$ and, accordingly, the *ratio of bucket* B as $ratio(B) = \|B\|_1 / (\|B\|_1 - \|B\|_\infty)$.

5.1. Our Key Algorithm Design Tools.

The example of Remark 4.1, 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 5.1, 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 is based on a reasonably involved exchange argument: we start with an optimal partition and then show how to exchange vectors among its buckets so that a new partition $\mathcal{P}'$ is obtained, that satisfies the desired properties.

Theorem 5.1. *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 impurity of a set B of i-pure vectors up to a $\log g$ factor times a lower bound on $I(B)$ as formalized by Proposition 5.2. Thus, in the light of Theorem 5.1, 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 $\chi^{2C}(\mathbf{v}) = (\|\mathbf{v}\|_\infty, \|\mathbf{v}\|_1 - \|\mathbf{v}\|_\infty)$ if $\|\mathbf{v}\|_\infty \geq \frac{1}{2}\|\mathbf{v}\|_1$ and $\chi^{2C}(\mathbf{v}) = (\|\mathbf{v}\|_1/2, \|\mathbf{v}\|_1/2)$, otherwise.

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

Proposition 5.2. *Fix $i \in [g]$ and let B be an i-pure bucket. It holds that $(1/2)I_2(B) \leq I(B) \leq 2I_2(B) + O(\log g)\sum_{\mathbf{v}\in B} I(\mathbf{v})$.*

Finally, our last tool is the following result from (Kurkoski & Yagi, 2014), here stated following our notation, that shows that PMWIP can be optimally solved when $g = 2$.

Theorem 5.3 ((Kurkoski & Yagi, 2014)). *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.*

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 5.3 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.

By Proposition 5.2 for an i-pure set of vectors, B, 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.

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

5.2. 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 5.1 for one that (roughly speaking) minimizes the impurity of its mixed bucket plus the sum of the 2-impurity of its pure buckets.

A pseudo-polytime algorithm. Given an instance of the PMWIP, let $S_j = \{\mathbf{v}|dom(\mathbf{v}) = j' \text{ for some } j' \leq j\}$ and $C = \sum_{\mathbf{v}\in V}\|\mathbf{v}\|_1$. For fixed $w, i \in [g], \ell \in [|V|], c \in [C], b \in [L]$ 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 For the buckets in $\mathcal{Q}^*(w, \ell, S_j, b, c) \setminus B^{\mathcal{Q}^*}$, the sum of the 2-impurities is minimum among the partitions for S_j into b buckets that satisfy the previous items.

The algorithm builds partitions $\mathcal{Q}^*(w, \ell, S_g, k, c)$ for all the different 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 5.1. 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 for the impurity of a partition $\mathcal{Q}^* = \mathcal{Q}^*(w^*, \ell^*, S_g, k, c^*)$ the same upper bound $O(\log^2 g)OPT(V, I)$ holds as for the impurity of $\mathcal{P}^*$ (yielding Theorem 5.4). The key observations are: (i) the impurity of the bucket $B^{\mathcal{Q}^*}$ of $\mathcal{Q}^*$ satisfies the same upper bound as the the impurity 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 $\mathcal{P}^* \setminus B^*_{mix}$. Hence, by Proposition 5.2, up to constant factors, the sum of the impurity of the buckets in $\mathcal{Q}^* \setminus B^{\mathcal{Q}^*}$ is at most the sum of the impurity of the buckets $\mathcal{P}^* \setminus B^*_{mix}$ plus $O(\log g) \sum_{\mathbf{v} \in V} I(\mathbf{v}) = O(\log g)OPT(V, I)$. Therefore, by the assumption $I(\mathcal{P}^*) = O(\log^2 g)OPT(V, I)$ we get the desired bound $I(\mathcal{Q}^*) = O(\log^2 g)OPT(V, I)$.

Now we show how to build the partitions $\mathcal{Q}^*(w, \ell, S_i, b, c)$. To simplify let us also assume, w.l.o.g., that $w = 1$. Let $\mathcal{Q}^* = \mathcal{Q}^*(1, \ell, S_i, b, c)$ 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 pure by definition. Note that $\mathcal{Q}^*$ is the partition with minimum $I_2^{pure}()$ among those that satisfy (a)-(d).

Let $V_j = \{\mathbf{v} | dom(\mathbf{v}) = j\}$ and $V_i(j)$ be the set of the j vectors of V_i of smallest ratio. In addition, 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 $V_i(j)$.

The 2-impurity of the pure buckets of $\mathcal{Q}^*(w, \ell, S_i, b, c)$, by property (e), satisfies the following recurrence when $i > 1$.

$$I_2^{pure}(\mathcal{Q}^*(1, \ell, S_i, b, c)) = \min_{\substack{0 \le j \le |V_i| \\ 0 \le b' < b}} \{ I_2(\mathcal{A}^{2C}(V_i \setminus V_i(j), b')) + I_2^{pure}(\mathcal{Q}^*(1, \ell, S_{i-1}, b - b', c - c_i(j))) \} \tag{8}$$

In fact, for each i and for some j the first j many i-dominant vectors (of smallest ratio) are in $B^{\mathcal{Q}^*}$ and the remaining i-dominant vectors are optimally partitioned in some $b' < b$ many i-pure buckets, where the optimality is with respect to their 2-impurity, which is given by algorithm $\mathcal{A}^{2C}$.

In the basis case, where $i = 1$, we have $I_2^{pure}(\mathcal{Q}^*(1, \ell, S_1, b, c)) = I_2(\mathcal{A}^{2C}(V_1 \setminus V_1(\ell), b - 1))$, if $c = c_1(\ell)$, and $I_2^{pure}(\mathcal{Q}^*(1, \ell, S_1, b, c)) = \infty$, otherwise.

For a fixed w and ℓ, our pseudo-polynomial algorithm employs a dynamic programming approach based on equation (8). It uses a list U_i to store the partitions $\mathcal{Q}^*(w, \ell, S_i, b, c)$ such that $I(\mathcal{Q}^*(w, \ell, S_i, b, c)) \neq \infty$. It first constructs the list U_1 using the basis case and then U_{i+1} from U_i using Equation (8). The partition with minimum impurity of type $\mathcal{Q}^*(w, \ell, S_k, L, c)$ is returned.

It is not hard to argue that the algorithm runs in polynomial time on $n = |V|$ and $C = \sum_{\mathbf{v} \in V} \|\mathbf{v}\|_1$. Then, from the approach of Section 3 (equation (5)) we have the following.

Theorem 5.4. *There exists a pseudo-polynomial time $O(\log^2(\min\{g, k\}))$-approximation algorithm for* PMWIP.

A polynomial time algorithm. The key idea to obtain a polytime algorithm is to employ a variation of the pseudo-polytime algorithm in which the number of partitions in the list U_i is controlled so that we can guarantee the existence of a polynomial number of them. For that, the interval $[0, \sum_{\mathbf{v} \in V} \|\mathbf{v}\|_1]$ is partitioned into $4k$ uniform subintervals and just the partition with the smallest impurity, among those that lie in the subinterval, is kept. The subinterval of a partition $\mathcal{Q}(w, \ell, S_i, b, c)$ is defined by its fifth parameter c.

The approximation relies on the fact that for every partition $\mathcal{P}$ built by the pseudo-polytime algorithm there exists a partition built by the poly-time algorithm that has impurity similar to that of $\mathcal{P}$. The proof has the same flavor of the one employed to show that the SUBSET SUM problem admits a FPTAS. As a result we have:

Theorem 5.5. *There is a polynomial time $O(\log^2(\min\{g, k\}))$ approximation algorithm for* PMWIP.

6. Innaproximability of PMWIP$_{Ent}$ and MTC$_{KL}$

We give the main ideas of our proof that PMWIP$_{Ent}$ is APX-Hard. Our strategy consists of reducing the c-gap problem, associated with the minimum vertex cover in cubic graphs, to a c'-gap problem for PMWIP$_{Ent}$. A c-gap problem for the former was established in (Chlebík & Chlebíková, 2006).

Let $G = (V, E)$ be a cubic graph for which it is hard to distinguish whether the minimum vertex cover has size at most k or larger than ck, where c is a constant larger than 1. We build a reduced instance $\mathcal{R}$ for PMWIP$_{Ent}$ as follows: we create a binary vector $\mathbf{v}_e$ of dimension $|V|$, for each edge $e = ij \in E$, in which the only components with value 1 are i and j. We note that every vector in the reduced instance has ℓ_1 norm 2. Moreover, the number of clusters in $\mathcal{R}$ is k.

We prove that: (a) if G has a vertex cover of size k then $\mathcal{R}$ has a partition with impurity at most $k' = (|E| - 2k)(6 + 3\log 3) + (3k - |E|)6$ and (b) if the minimum vertex cover in G has size at least ck then every partition of size k in $\mathcal{R}$ has impurity at least $c'k'$ for some constant $c' > 1$

The correctness of item (a) relies on the following structural property of cubic graphs: if a cubic graph $G = (V, E)$ has a minimal vertex cover with k vertexes then it is possible to decompose G into k stars such that each of them has either 2 or 3 edges. The value of k', given in the previous paragraph, is exactly the impurity of the clustering induced by this set of k stars. The correctness of item (b) is based on the following facts: (i) k' is a lower bound on the impurity of any k-clustering for any set of $|E|$ binary vectors such that all of them have ℓ_1 norm 2 and (ii) if the minimum vertex cover of G has size at least $k \cdot c$ then any k-clustering of $\mathcal{R}$ must have a non negligible number of clusters that do not correspond to stars with 2 or 3 edges so that its impurity will be at least $k'c'$.

The same arguments 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$_{KL}$. Then, we have

Theorem 6.1. *The* PMWIP$_{Ent}$ *is APX-Hard even for the case where all vector have the same ℓ_1 norm. Hence,* MTC$_{KL}$ *is APX-Hard, too.*

7. New Fast Method for Information Clustering

Although the focus of our research is mainly theoretical, we also designed RATIO-GREEDY, a fast and practical algorithm that relies on the results developed so far.

The RATIO-GREEDY algorithm. If $k \leq g$, RATIO-GREEDY runs the $\mathcal{A}^{Dom}$, the dominance algorithm presented in Section 4. Thus, we focus on the case $k > g$. Let V_i be the set of i-dominant vectors and let L_i be the list obtained by sorting the vectors in V_i according to their ratios as defined in Section 5. For the following explanation it will be convenient to think of L_i as a list of adjacent clusters that initially contains $|V_i|$ unitary clusters.

RATIO-GREEDY predefines the number of clusters t_i that will be available for the i-dominant vectors so that $\sum_{i=1}^{g} t_i = k$. It heuristically set $t_i = kI_{Ent}(V_i)/\sum_{j=1}^{g} I_{Ent}(V_j)$.

Then, for each i, RATIO-GREEDY greedily reduces the number of clusters in L_i from $|V_i|$ to t_i by iteratively selecting two adjacent clusters in the current list and replacing them with their union so that a list with one less cluster is obtained. The pair of adjacent clusters 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 $loss(C, C') = I_{Ent}(C \cup C') - I_{Ent}(C) - I_{Ent}(C')$.

RATIO-GREEDY can be implemented to run in $O(n \log n + ng)$ time, exploiting a binary heap to select the adjacent clusters in L_i whose merge incurs the minimum $loss$. Note that the impurity of the partition obtained by RATIO-GREEDY is no worse than that obtained by $\mathcal{A}^{Dom}$ due to the superadditivity of I_{Ent}, thus it inherits its approximation guarantees.

Experiments. We compared RATIO-GREEDY with DIVISIVE CLUSTERING (DC for short), an adaptation of the k-means method proposed in (Dhillon et al., 2003) to solve PMWIP$_{Ent}$. The key difference between them is that DC employs KL-divergence rather than Euclidean squared distance. When $k > g$, the initialization of DC resembles $\mathcal{A}^{Dom}$ since it consists of splitting the vectors in V_i among k/g clusters. When $k \leq g$ we initialize DC using $\mathcal{A}^{Dom}$.

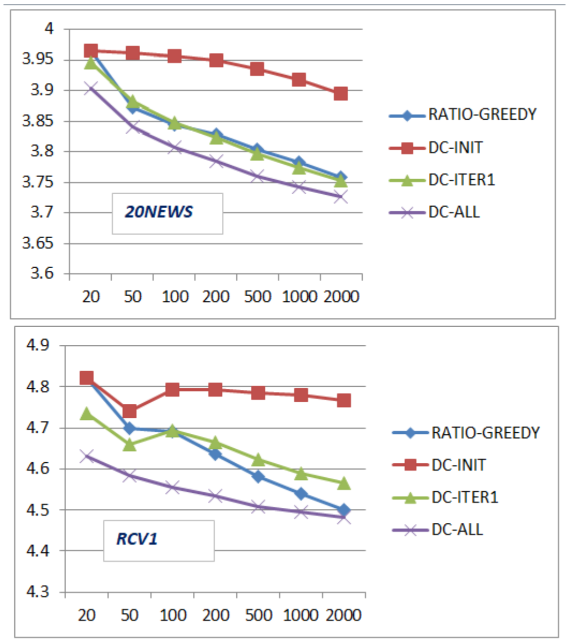

Figure 1. Impurities (vertical axis) of the partitions obtained by RATIO-GREEDY and DIVISIVE-CLUSTERING for different values of k (horizontal axis).

We tested these methods on clustering 51.480 words from the 20NEWS corpus and 170.946 words from RCV1 corpus, according to their distributions w.r.t. 20 and 103 different classes respectively. The distribution vectors associated with the words are built according to the methodology employed in (Dhillon et al., 2003) to address text classification tasks. 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 DC: right after its initialization, after its first iteration and at the end. 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. As an example for RCV1, with $k = 2000$, it is 55 times faster than a *single iteration* of DC. Moreover, after 5 iterations of DC, RATIO-GREEDY still had a partition with lower impurity.

For additional details of the experiments see the supplementary material. Code and datasets are available in (Murtinho, 2019)

Acknowledgements

We thanks to Daniel Mesejo for pointing out reference (Lucic et al., 2016). The work of the second author is partially supported by CNPq under grant 307572/2017-0 and by FAPERJ, grant Cientista do Nosso Estado E-26/202.823/2018.

References

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

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

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

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

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

Baker, L. D. and McCallum, A. K. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 96–103. ACM Press, 1998. ISBN 1-58113-015-5.

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

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

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

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

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

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

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

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

Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, September 1999. ISSN 0360-0300.

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

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

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

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

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

Li, T., Ma, S., and Ogihara, M. Entropy-based criterion in categorical clustering. In Brodley, C. E. (ed.), *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.

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

Murtinho, L. Codes and datasets, 2019. URL https://github.com/lmurtinho/RatioGreedyClustering/tree/ICML_submission.

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

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

Slonim, N. and Tishby, N. Agglomerative information bottleneck. In Solla, S. A., Leen, T. K., and Müller, K.-R. (eds.), *NIPS*, pp. 617–623. The MIT Press, 1999. ISBN 0-262-19450-3.

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

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