Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 346 (2019) 567–576

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Minimization of Gini Impurity: NP-completeness and Approximation Algorithm via Connections with the k-means Problem

Eduardo Laber,[1] Lucas Murtinho[2]

PUC-Rio
Rio de Janeiro, Brazil

Abstract

The Gini impurity is a very popular criterion to select attributes during decision trees construction. In the problem of finding a partition with minimum weighted Gini impurity ($PMWGP$), the one faced during the construction of decision trees, a set of vectors must be partitioned into k different clusters such that the partition's overall Gini impurity is minimized.
We show that $PMWGP$ is APX-hard for arbitrary k and admits a randomized PTAS when the number of clusters is fixed. These results significantly improve the current knowledge on the problem. The key idea to obtain these results is to explore connections between $PMWGP$ and the geometric k-means clustering problem.

Keywords: Impurity, Gini, k-means, NP-completeness, Approximation.

1 Introduction

Decision Trees and Random Forests are among the most popular methods for classification tasks. It is widely known that decision trees, especially small ones, are easy to interpret, while random forests usually yield more stable/accurate classifications.

A key decision during the construction of these structures is the selection of the attribute that is used for branching at each node. The standard approach for this selection is to evaluate the ability of each attribute to generate "pure" partitions, that is, partitions in which each branch is very homogeneous with respect to the class distribution of its examples. To measure how impure each branch is, impurity measures are often employed.

[1] Email: laber@inf.puc-rio.br
[2] Email: lucas.murtinho@gmail.com

https://doi.org/10.1016/j.entcs.2019.08.050

An impurity measure maps a vector $\mathbf{u} = (u_1, \ldots, u_d)$, counting how many examples of each class we have in a node (branch), into a non-negative scalar [3]. Arguably, one of the most classical impurity measures is the *Gini* impurity i_{Gini}, which measures the probability of misclassification when an object is assigned to class i with probability $\frac{u_i}{\|\mathbf{u}\|}$. It is defined as:

$$i_{Gini}(\mathbf{u}) = \sum_{i=1}^{d} \frac{u_i}{\|\mathbf{u}\|_1} \left(1 - \frac{u_i}{\|\mathbf{u}\|_1}\right).$$

The Gini impurity is used in the CART package for classification and regression decision trees [3]. It is therefore important to understand the complexity of the problem of finding optimal partitions w.r.t. Gini impurity, and also to build heuristics and algorithms for this task.

Problem Definition. For a vector $\mathbf{v}$ where all components are non-negative, the weighted Gini impurity $Gini(\mathbf{v})$ is defined as $Gini(\mathbf{v}) = \|\mathbf{v}\|_1 \cdot i_{Gini}(\mathbf{v})$. Let A be a nominal attribute that may take n possible values $a_1, \ldots, a_n$. The k-ary Partition with Minimum Weighted Gini Problem (k-*PMWGP*) can be described abstractly as follows. We are given a collection of n non-null vectors $V \subset \mathbb{Z}^d$, where the i-th component of the j-th vector counts the number of examples in class i for which the attribute A has value a_j. The goal is to find a partition $\mathcal{P}$ of V into k groups so as to minimize the sum of the weighted Gini impurities

$$Gini(\mathcal{P}) = \sum_{i=1}^{k} Gini\left(\sum_{\mathbf{v} \in V_i} \mathbf{v}\right). \tag{1}$$

A problem that is equivalent to the above one from the perspective of optimality (but different from the perspective of approximation) is finding the partition $\mathcal{P}$ of V into k groups that minimizes

$$Gini(\mathcal{P}) - \sum_{\mathbf{v} \in V} Gini(\mathbf{v}). \tag{2}$$

Using concavity properties of Gini, one can prove that the above expression is always non-negative. An α-approximation with respect to goal (2) implies an α-approximation with respect to goal (1), but the converse is not necessarily true, so that approximations with respect to goal (2) are stronger [7].

In order to properly explain our work we shall recall the definition of the classic **k-means clustering.** In the geometric k-means problem we are given a set of vectors $V \subset \mathbb{R}^d$ and the goal is to find a partition $\mathcal{P}$ of V into k groups $V_1, \ldots, V_k$ and a set of k centers $\mathbf{c}_1, \ldots, \mathbf{c}_k$ in $\mathbb{R}^d$ such that

$$Cost_{KM}(\mathcal{P}) = \sum_{i=1}^{k} \sum_{\mathbf{v} \in V_i} \|\mathbf{v} - \mathbf{c}_i\|_2^2$$

[3] In the original definition an impurity measure maps a vector of probabilities into a non-negative scalar.

is minimized.

It is well known that if U is a set of vectors then the vector $\mathbf{c}$ for which $\sum_{\mathbf{v}\in U} \|(\mathbf{v}-\mathbf{c})\|_2^2$ is minimum is the centroid of U, that is, $\mathbf{c} = (\sum_{\mathbf{v}\in U} \mathbf{v})/|U|$.

Our Results. We show that *PMWGP* is APX-hard for arbitrary k and admits a randomized PTAS when the number k of clusters is fixed. These results significantly improve the current knowledge on the problem.

The key idea to obtain these results is to explore connections between *PMWGP* and the k-means clustering problem. In fact, the hardness of *PMWGP* relies on a reduction from the vertex cover problem in triangle-free graphs to the k-means clustering problem due to Awasthi et al. [2]. Furthermore, our randomized PTAS for *PMWGP* is a simple adaptation of the randomized PTAS for the k-means problem presented by Kumar et al. [9] and further explored by Ackermann et al. [1].

Related Work. There have been theoretical investigations on methods to compute the best split efficiently for impurity measures such as the Gini impurity. For $d = k = 2$, Breiman [3] presented a simple algorithm that finds the best binary partition in $\mathcal{O}(n \log n)$ time for impurity measures in a certain class that includes Gini. The correctness of this algorithm relies on a theorem, also proved in [3], which is generalized for arbitrary d and k in Chou [5], Burshtein et al. [4], and Coppersmith et al. [7]. Basically, these theorems provide necessary conditions for partitions with minimum impurity and can be used to restrict the set of partitions that need to be considered. A connection between Gini and the squared ℓ_2 distance employed by k-means, that we explore here, is mentioned in the appendix of Chou [5].

The results presented here belong to the research project of the first author and his collaborators, whose goal is to obtain a better understanding, from the perspective of approximation algorithms, of clustering based on impurity measures/Bregman divergences. In [10], new splitting procedures that provably achieve near-optimal impurity for binary partitions based on a family of measures that include the Gini impurity as well as the Entropy impurity are presented. In [6], approximation algorithms for the same family of measures in the more general problem of k-ary partitions are given.

For the Euclidean k-means problem a vast literature is available and the problem is well understood from the perspective of approximation algorithms in the sense that the gap between the best available approximation factors and the thresholds given by the hardness results is small (see [2] and the references therein).

Paper Organization. In Section 2, we present the connections between the Partition with Minimum Weighted Gini Problem (*PMWGP*) and the k-means clustering problem. In addition, we use these connections to prove the hardness of *PMWGP*. Section 3 analyzes an algorithm that, for a fixed k and with constant probability, provides a $(1+\epsilon)$-approximation for *PMWGP* in polynomial time. We discuss the results and present further research directions in the conclusions section.

2 Connections between Gini minimization and k-means clustering

In this section we argue that the following connections between k-*PMWGP* and the k-means problem hold:

Proposition 2.1 *Let V be an instance of k-means in which all vectors have the same ℓ_1 norm. If $\mathcal{P}$ is an optimal partition for instance V of k-means, then $\mathcal{P}$ is also an optimal partition for instance V of k-PMWGP.*

Proposition 2.2 *There exists a pseudo-polynomial time reduction from k-PMWGP to the geometric k-means problem.*

The key observation for establishing both propositions is the following lemma.

Lemma 2.3 *Let X be a set of n vectors, all of them with ℓ_1 norm equal to L. Then,*

$$Gini\left(\sum_{\mathbf{v}\in X}\mathbf{v}\right) - \sum_{\mathbf{v}\in X} Gini(\mathbf{v}) = \frac{1}{L}\times\left(\sum_{\mathbf{v}\in X}\|\mathbf{v}-\mathbf{c}\|_2^2\right),$$

where $\mathbf{c}$ is the centroid of the set of vectors in X.

Proof. Let $\mathbf{u} = \sum_{\mathbf{v}\in X}\mathbf{v}$ and d be the dimension of the vectors in X. We have that

$$\begin{aligned} Gini(\mathbf{u}) - \sum_{\mathbf{v}\in X} Gini(\mathbf{v}) &= \|\mathbf{u}\|_1 \sum_{i=1}^{d}\left(\frac{u_i}{\|\mathbf{u}\|_1}\right)\left(1-\frac{u_i}{\|\mathbf{u}\|_1}\right) \\ &\quad - \sum_{\mathbf{v}\in X}\|\mathbf{v}\|_1\sum_{i=1}^{d}\left(\frac{v_i}{\|\mathbf{v}\|_1}\right)\left(1-\frac{v_i}{\|\mathbf{v}\|_1}\right) \\ &= \sum_{i=1}^{d}\left(u_i - \frac{(u_i)^2}{L\times n}\right) - \sum_{i=1}^{d}\sum_{\mathbf{v}\in X}\left(v_i - \frac{(v_i)^2}{L}\right). \end{aligned}$$

On the other hand,

$$\sum_{\mathbf{v}\in X}\|\mathbf{v}-\mathbf{c}\|_2^2 = \sum_{i=1}^{d}\sum_{\mathbf{v}\in X}(v_i-c_i)^2.$$

Thus, it suffices to show that, for any i,

$$u_i - \frac{(u_i)^2}{L\times n} - \sum_{\mathbf{v}\in X}\left(v_i - \frac{(v_i)^2}{L}\right) = \frac{1}{L}\left(\sum_{\mathbf{v}\in X}(v_i-c_i)^2\right). \tag{3}$$

The left side of (3) is equal to

$$u_i - \frac{(u_i)^2}{L\times n} - u_i + \frac{\sum_{\mathbf{v}\in X}(v_i)^2}{L} = \frac{1}{L}\times\left(\sum_{\mathbf{v}\in X}(v_i)^2 - \frac{(u_i)^2}{n}\right).$$

Moreover, the right side of (3) is equal to

$$\frac{1}{L} \times \left(\sum_{\mathbf{v} \in X} (v_i)^2 - 2c_i \sum_{\mathbf{v} \in X} v_i + \sum_{\mathbf{v} \in X} (c_i)^2 \right)$$

$$= \frac{1}{L} \times \left(\sum_{\mathbf{v} \in X} (v_i)^2 - 2\frac{(u_i)^2}{n} + n\frac{(u_i)^2}{n^2} \right) = \frac{1}{L} \times \left(\sum_{\mathbf{v} \in X} (v_i)^2 - \frac{(u_i)^2}{n} \right),$$

which establishes the lemma. □

Proposition 2.1 is a direct consequence of Lemma 2.3, since it implies that, for all k-partitions $\mathcal{P}$ of V,

$$Gini(\mathcal{P}) = \frac{1}{L} \cdot Cost_{KM}(\mathcal{P}) + \sum_{\mathbf{v} \in V} Gini(\mathbf{v}).$$

With regard to Proposition 2.2, let V be an instance of *k-PMWGP* and let V' be the instance of k-means obtained from V as follows: for each vector $\mathbf{v} \in V$, we add to the instance set V' exactly $\|\mathbf{v}\|_1$ copies of the vector $\mathbf{v}' = \mathbf{v}/\|\mathbf{v}\|_1$. Using Lemma 2.3 and also the fact that in any optimal solution for k-means identical vectors are in the same partition, we conclude that the optimal value of V and V' differ by exactly $\sum_{\mathbf{v} \in V} Gini(\mathbf{v})$. Note that instance V' is obtained from V in pseudo-polynomial time.

From Proposition 2.1 and the hardness of geometric k-means established in [2] we obtain:

Theorem 2.4 *The Partition with Minimum Weighted Gini Problem (PMWGP) is NP-complete with respect to goal* (1) *and APX-hard with respect to goal* (2)*.*

Proof. Theorem 1.1 of [2] states that there is a constant ϵ such that it is NP-hard to approximate the k-means problem to a factor better than $(1+\epsilon)$. In the instance used to prove the theorem in [2], all vectors have ℓ_1 norm of 2; from our Lemma 2.3, it follows that the goal (2) and the objective function for k-means differ by a factor of exactly 2. The NP-completeness of goal (1) follows because goals (1) and (2) differ by an additive constant. □

3 Approximating the optimal Gini partition

The reduction given by Proposition 2.2 allows us to obtain new algorithms for *k-PMWGP* with provable approximation factors. As an example, we discuss how to obtain a randomized PTAS for *k-PMWGP* with respect to the objective function (2) when k is fixed. To do so, we run over instance V' the randomized PTAS for the k-means problem proposed by Kumar et al. [9]. Algorithm 1 (Cluster) is a slightly modified version of the algorithm as presented in Figure 1 of [1].

In what follows we explain how Algorithm 1 works and how we adapt it to our purposes. However, we do not detail the analysis of its approximation, since it is not necessary to establish our result. We refer the reader to [1] for a complete

presentation of the PTAS proposed in [9]. This presentation includes a simplified proof of its approximation factor as well as a discussion of how it generalizes to other objective functions.

Algorithm 1 CLUSTER$(R, l, \tilde{C})$

Input:
R: set of remaining input points
l: number of medians to be found
$\tilde{C}$: set of medians already found
Procedure:

1: **if** $l = 0$ **then** return $\tilde{C}$
2: **else**
3: **if** $l \geq |R|$ **then** return $\tilde{C} \cup R$
4: **else**
5: $C^{(\tilde{c})} = \{\}$
6: */* sampling phase */*
7: sample a multiset S of size $\frac{2}{\alpha\gamma\delta}$ from R
8: $C \leftarrow \left\{ \frac{\sum_{\mathbf{v} \in S'} \mathbf{v}}{|S'|} \mid S' \subset S,\ |S'| = \frac{1}{\gamma\delta} \right\}$ *//set of median candidates*
9: **for all** $\tilde{c} \in C$ **do**
10: $C^{(\tilde{c})} \leftarrow C^{(\tilde{c})} \cup$ CLUSTER$(R, l-1, \tilde{C} \cup \{\tilde{c}\})$
11: */* pruning phase */*
12: let N be the set with the $\frac{1}{2}|R|$ minimal points $p \in R$ w.r.t. $\min_{\tilde{c} \in \tilde{C}} ||p - \tilde{c}||_2^2$
13: $C^{(\tilde{c})} \leftarrow C^{(\tilde{c})} \cup$ CLUSTER$(R \setminus N, l, \tilde{C})$
14: **return** $C \in C^{(\tilde{c})}$ with minimum cost

CLUSTER takes as input the set R of remaining objects to be clustered; the number l of medians yet to be found; and the set $\tilde{C}$ of medians already found. (Initially, $R = V$, $l = k$ and $\tilde{C} = \emptyset$.) In addition, it employs 3 parameters: γ, which is used to control the approximation factor; δ, which is used to control the probability of returning a good cluster in terms of approximation; and α, a positive constant that impacts both the running time and the approximation factor. CLUSTER returns the set $\tilde{C}$, containing k medians. The points of V can then be clustered according to their closest points in the set $\tilde{C}$.

If there are no medians to be found ($l = 0$), then the algorithm returns the set $\tilde{C}$. In another base case, when $l \geq |R|$, each of the points in R becomes a new median. Otherwise, the algorithm considers strategies (i) and (ii), presented below, to cluster the points in R, and returns the best clustering among those built by these strategies.

(i) The algorithm builds a set C containing the centroids of all subsets of S with size $1/\gamma\delta$, where S is a random multiset of R with size $2/\alpha\gamma\delta$. Each centroid $\tilde{c} \in C$ is added separately to $\tilde{C}$, and the algorithm performs $|C|$ recursive calls with parameters $(R, l-1, \tilde{C} \cup \tilde{c})$. This strategy corresponds to lines 7-10.

(ii) The algorithm builds a set N containing the $|R|/2$ points in R that are closest to the medians in $\tilde{C}$. Then, the procedure is recursively called with parameters $(R \setminus N, l, \tilde{C})$. This strategy corresponds to lines 12-13.

Therefore, we can define the number of calls made by CLUSTER as:

$$T(|R|, l) = \begin{cases} 1 & \text{if } l = 0 \text{ or } |R| \leq l \\ c \times T(|R|, l-1) + T(|R|/2, l) + 1 & \text{otherwise} \end{cases} \tag{4}$$

where c is the cardinality of the set C. An important fact is that c is constant with respect to $|V|$ and k, since it only depends on γ, δ and α. In fact, $c < 2^{|S|} = 2^{\frac{2}{\alpha\gamma\delta}}$.

Theorem 2.8 of [1] shows that CLUSTER executes at most $n2^{O(k/\gamma\delta\cdot\log(1/(\gamma\delta\alpha)))}$ arithmetic operations. In addition, Theorem 2.5 of [1] shows that, for $\alpha < \frac{1}{4k}$,

$$Pr[Cost(\tilde{C}) \leq (1 + 8\alpha k^2)(1+\gamma)OPT] \geq \left(\frac{1-\delta}{5}\right)^k, \tag{5}$$

where $Cost(\tilde{C})$ is the cost of a clustering induced by the points in $\tilde{C}$ and OPT is the cost of the optimal solution.

Algorithm 1 for the minimization of Gini. By applying CLUSTER to the instance V', obtained via Proposition 2.2, we find with probability at least $((1-\delta)/5)^k$ a partition whose Gini is not larger than $(1+\epsilon)$ times the Gini of an optimal partition, where $\epsilon = 8\alpha k^2(1+\gamma)$. Since the size of V' is pseudopolynomial on the size of V, CLUSTER$(V', k, \{\})$ would have an exponential running time w.r.t. $n = |V|$ in the worst case. This would be the case, however, if V' were represented as a simple list of vectors; we prove below that, with a more compact representation, the running time of CLUSTER$(V', k, \{\})$ remains polynomial on n.

The first step is to build V' from V in polynomial time. We do this by keeping each distinct vector $\mathbf{v}' = \mathbf{v}/\|\mathbf{v}\|_1$ and its multiplicity $||\mathbf{v}||_1$ rather than all the $\sum_{\mathbf{v}\in V} \|\mathbf{v}\|_1$ vectors of instance V'. Next, we need to establish an upper bound for the running time of each call to CLUSTER, which we do in Lemma 3.1 below. Finally, in Lemma 3.2, we prove that the total number of recursive calls performed by CLUSTER$(V', k, \{\})$ is polynomial on n.

Lemma 3.1 *Given a fixed k, a single call to* CLUSTER$(V', l, \tilde{C})$ *performs $\mathcal{O}(n \lg n)$ operations, where $n = |V|$.*

Proof. Let $W = |V'| = \sum_{\mathbf{v}\in V} ||\mathbf{v}||_1$. In the base case when $l = 0$, the algorithm returns the set of medians found, at constant cost. In the base case when $W \leq l$, the algorithm assigns the W remaining points as medians; this would take $\mathcal{O}(W)$ operations, but we are keeping track of the n different vectors in V' and their multiplicities, so assigning them as medians is $\mathcal{O}(n)$.

It remains to evaluate the complexity of calls when no base case has been reached. The only steps in CLUSTER that depend on the size of the input are steps 7 and 12,

and we analyze them for input V' as follows:

(i) Step 7 samples a constant number of vectors from V'. Since the W vectors in V' are being represented as n vectors along with their multiplicities, we can sample these n vectors in proportion to their quantities, and no additional cost is incurred.

(ii) Step 12 computes the $W/2$ closest vectors to a given set of medians $\tilde{C}$. This is achieved in [1] by finding the median element of V' according to their distances to the medians, taking $\mathcal{O}(W)$ operations. Instead, we can order the n unique vectors in V' according to their minimal distance to $\tilde{C}$, and then select the $W/2$ closest vectors by checking the multiplicities of the vectors in V'. This will take $\mathcal{O}(n \lg n)$ operations.

Therefore, the number of operations of any call of CLUSTER with V' as input is $\mathcal{O}(n \lg n)$. □

Lemma 3.2 *For the recurrence presented in (4), $T(W,k) \leq c^{k+1}(\lg((k+1)W))^k - c^k$, where c is a constant smaller than $2^{\frac{2}{\alpha\gamma\delta}}$.*

Proof. For the base cases mentioned above:

- $k = 0$: $T(W,0) = 1 \leq c(\lg W)^0 - c^0$ which holds as long as $c \geq 2$;
- $k \geq W \geq 1$: $T(W,k) = 1 \leq c^{k+1}(\lg((k+1)W))^k - c^k$ which holds because $\frac{1}{c^k} + 1 \leq 2 \leq c \leq c(\lg((k+1)W))^k$.

Let $W > k \geq 1$ and assume that the lemma holds for all W', k' with $W' < W$ or $k' < k$. By induction, we can use recurrence (4) to show that:

$$\begin{aligned} T(W,k) &\leq c(c^k(\lg(kW))^{k-1} - c^{k-1}) + c^{k+1}\left(\lg\frac{(k+1)W}{2}\right)^k - c^k + 1 \\ &= c^{k+1}\left(\left(\lg(kW)\right)^{k-1} + \left(\lg\frac{(k+1)W}{2}\right)^k\right) - 2c^k + 1. \end{aligned} \tag{6}$$

We must therefore show that (6) $\leq c^{k+1}(\lg((k+1)W))^k - c^k$. Since $1 \leq c^k$ if $c \geq 1$, it suffices to prove

$$\begin{aligned} \left(\lg(kW)\right)^{k-1} + \left(\lg\frac{(k+1)W}{2}\right)^k &\leq \left(\lg((k+1)W)\right)^{k-1} + \left(\lg\frac{(k+1)W}{2}\right)^k \\ &\leq (\lg((k+1)W))^k \end{aligned}$$

which we can do by showing that

$$1 + \frac{\left(\lg \frac{(k+1)W}{2}\right)^k}{(\lg((k+1)W))^{k-1}} \leq \lg((k+1)W) \quad \Longrightarrow \quad \frac{\left(\lg \frac{(k+1)W}{2}\right)^k}{(\lg((k+1)W))^{k-1}} \leq \lg \frac{(k+1)W}{2}$$
$$\frac{\left(\lg \frac{(k+1)W}{2}\right)^{k-1}}{(\lg((k+1)W))^{k-1}} \leq 1. \tag{7}$$

Since (7) is always true, the lemma holds. □

Theorem 3.3 *For any $\epsilon, \delta > 0$ and for a fixed k,* Cluster *provides, in polynomial time and with probability $\geq 1 - \delta$, a $(1+\epsilon)$-approximation to $k-PMWGP$.*

Proof. Directly from Lemmas 3.1 and 3.2: since, for a fixed k, there are at most $\mathcal{O}(n^k)$ calls to Algorithm 1, each performing at most $\mathcal{O}(n \lg n)$ operations, the overall running time of Cluster$(V', k, \{\})$ is $\mathcal{O}(n^{k+1} \lg n)$. □

4 Conclusion

We show in this paper how the connections between minimizing the Gini impurity of a partition and the geometric k-means problem provide new tools to better understand and work with the former problem. In particular, these connections allow us to establish hardness results for *PMWGP* and to find a polynomial-time, probabilistic $(1+\epsilon)$-approximation algorithm for the problem when k is fixed.

One natural question is whether other algorithms can be adapted to the *PMWGP* via the connection presented here. Since the size of a k-means instance built from a *PMWGP* instance can be pseudopolynomial on the size of the *PMWGP* instance, it may be the case that some algorithms do not remain polynomial when used through this connection, or that some known results for the k-means problem do not hold for the *PMWGP*. As an example, the algorithm of [8], a constant-approximation algorithm for the k-means problem, relies on the existence of a set of ϵ-centroids of cardinality $\mathcal{O}(n)$. It is not yet clear whether the pseudopolynomial transformation presented above would allow the *PMWGP* to be (approximately) solved in polynomial time by this algorithm.

References

[1] Ackermann, M. R., J. Blömer and C. Sohler, *Clustering for metric and nonmetric distance measures*, ACM Trans. Algorithms **6** (2010), pp. 59:1–59:26.
URL http://doi.acm.org/10.1145/1824777.1824779

[2] Awasthi, P., M. Charikar, R. Krishnaswamy and A. K. Sinop, *The hardness of approximation of Euclidean k-means*, in: L. Arge and J. Pach, editors, *31st International Symposium on Computational Geometry (SoCG 2015)*, Leibniz International Proceedings in Informatics (LIPIcs) **34** (2015), pp. 754–767.
URL http://drops.dagstuhl.de/opus/volltexte/2015/5117

[3] Breiman, L., "Classification and regression trees," Routledge, 2017.
URL https://www.taylorfrancis.com/books/9781351460491

[4] Burshtein, D., V. Della Pietra, D. Kanevsky, A. Nadas et al., *Minimum impurity partitions*, The Annals of Statistics **20** (1992), pp. 1637–1646.
URL https://projecteuclid.org/euclid.aos/1176348789

[5] Chou, P. A., *Optimal partitioning for classification and regression trees*, IEEE Transactions on Pattern Analysis & Machine Intelligence (1991), pp. 340–354.
URL https://www.computer.org/csdl/trans/tp/1991/04/i0340.pdf

[6] Cicalese, F. and E. Laber, *Approximation algorithms for clustering via weighted impurity measures*, arXiv preprint arXiv:1807.05241 (2018).
URL https://arxiv.org/abs/1807.05241

[7] Coppersmith, D., S. J. Hong and J. R. Hosking, *Partitioning nominal attributes in decision trees*, Data Mining and Knowledge Discovery **3** (1999), pp. 197–217.
URL https://link.springer.com/article/10.1023/A:1009869804967

[8] Kanungo, T., D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, *A local search approximation algorithm for k-means clustering*, Computational Geometry **28** (2004), pp. 89–112.
URL https://www.sciencedirect.com/science/article/pii/S0925772104000215

[9] Kumar, A., Y. Sabharwal and S. Sen, *A simple linear time $(1+\epsilon)$-approximation algorithm for k-means clustering in any dimensions*, in: *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, IEEE, 2004, pp. 454–462.
URL https://ieeexplore.ieee.org/abstract/document/1366265

[10] Laber, E. S., M. Molinaro and F. A. M. Pereira, *Binary partitions with approximate minimum impurity*, in: *International Conference on Machine Learning*, 2018, pp. 2860–2868.
URL http://proceedings.mlr.press/v80/laber18a.html