Sample-and-Search: An Effective Algorithm for Learning-Augmented k-Median Clustering in High dimensions

Kangke Cheng[1], Shihong Song[2], Guanlin Mo[1], Hu Ding[1*]

[1]University of Science and Technology of China, Hefei, China
[2]School of Informatics, University of Edinburgh, Edinburgh, UK
ke314159@mail.ustc.edu.cn, S.Song-29@sms.ed.ac.uk, moguanlin@mail.ustc.edu.cn, huding@ustc.edu.cn

Abstract

In this paper, we investigate the *learning-augmented k-median clustering* problem, which aims to improve the performance of traditional clustering algorithms by preprocessing the point set with a predictor of error rate $\alpha \in [0, 1)$. This preprocessing step assigns potential labels to the points before clustering. We introduce an algorithm for this problem based on a simple yet effective sampling method, which substantially improves upon the time complexities of existing algorithms. Moreover, we mitigate their exponential dependency on the dimensionality of the Euclidean space. Lastly, we conduct experiments to compare our method with several state-of-the-art learning-augmented k-median clustering methods. The experimental results suggest that our proposed approach can significantly reduce the computational complexity in practice, while achieving a lower clustering cost.

Code — https://github.com/KangkeCheng/Learning-Augmented-k-Median-Sample-and-Search

Extended version — https://github.com/KangkeCheng/Sample-and-Search-Full-Version

1 Introduction

As the core topics in unsupervised learning, *k-median* and *k-means* clusterings are widely applied to numerous fields, like bioinformatics (Kiselev et al. 2017), computer vision (Caron et al. 2020), and social network (Ghaffari, Mosavi, and Shamshirband 2021). The primary goal of these center-based clustering problems is to partition a set of unlabeled data points into multiple clusters, such that data points within the same cluster are similar to each other (under some metric), while data points in different clusters exhibit significant dissimilarity.

k-means problem seeks to find k centers that minimize the sum of squared Euclidean distances from each point to its nearest center. Formally, the goal is to minimize $\sum_{x\in X} \min_{c\in C} \|x - c\|_2^2$, where X is the input dataset and C is the set of k centers. Despite its popularity, k-means is known to be sensitive to outliers and noise, as the squared distance objective increases the impact of extreme values quadratically. In contrast, the k-median problem minimizes the sum of Euclidean distances: $\sum_{x\in X} \min_{c\in C} \|x - c\|_2$, which provides greater robustness to outliers and heavy-tailed distributions. Thus k-median clustering is preferable in many practical applications, especially when data is noisy. Therefore, our work focuses on the k-median setting, aiming to retain its robustness advantages while addressing the algorithmic challenges through the proposed framework.

Learning-Augmented algorithms. A central challenge in the field of algorithm design lies in simultaneously reducing algorithmic time complexity while maintaining a reliable approximation ratio. The proliferation of large-scale data and the advancement of machine learning bring the opportunity to obtain valuable prior knowledge for many classical algorithmic problems. To overcome the often pessimistic bounds of traditional worst-case analysis, the theoretical computer science community has introduced *learning-augmented* algorithms (Hsu et al. 2018; Antoniadis et al. 2020; Dinitz et al. 2021; Lykouris and Vassilvitskii 2021; Mitzenmacher and Vassilvitskii 2022)—a new paradigm that falls under the umbrella of "Beyond Worst-Case Analysis" (Roughgarden 2021). The core idea is to design algorithms that can harness auxiliary information, typically from a machine-learned model, to enhance their performance.

For learning-augmented k-means and k-median clustering, Gamlath et al. (2022) explored noisy labels, achieving $(1+O(\epsilon))$-approximation for balanced adversarial noise and $O(1)$-approximation for stochastic noise models. Here, the approximation ratio is a measure of the solution's quality, defined as the ratio of the cost of the algorithm's solution to the cost of the optimal solution. Ergun et al. (2022) developed a learning-augmented framework where data points are augmented with predicted labels, quantified by an error rate $\alpha \in [0, 1)$. They proposed a randomized algorithm for k-means problem that achieves a $(1 + 20\alpha)$-approximation under some specific constraints on α and cluster size in $O(nd \log n)$ time, where n denotes the number of points to be clustered and d denotes the dimension of the space. They also proposed an algorithm for the k-median problem that, under the condition that $\alpha = \tilde{O}\left(\frac{1}{k}\right)$, achieves an $\tilde{O}((k\alpha)^{1/4})$-approximation.

Nguyen, Chaturvedi, and Nguyen (2023) further improved these algorithms. Their k-means algorithm directly

*Corresponding author

Methods	Approximation Ratio	Label Error Range	Time Complexity
Ergun et al. (2022)	$1 + O((k\alpha)^{1/4})$	$\tilde{O}(\frac{1}{k})$	$O(nd \log^3 n + \text{poly}(k, \log n))$
Nguyen, Chaturvedi, and Nguyen (2023)	$1 + \frac{7\alpha + 10\alpha^2 - 10\alpha^3}{(1-\alpha)(1-2\alpha)}$	$[0, 1/2)$	$O(\frac{1}{1-2\alpha} nd \log^3 n \log^2 \frac{k}{\delta})$
Huang et al. (2025)	$1 + \frac{(6+\epsilon)\alpha - 4\alpha^2}{(1-\alpha)(1-2\alpha)}$	$[0, 1/2)$	$O(nd \log(kd) \log(n\Delta) \cdot (\frac{\sqrt{d}}{\alpha\epsilon})^{O(d)})$
Sample-and-Search (ours)	$1 + \frac{(6+\epsilon)\alpha - 4\alpha^2}{(1-\alpha)(1-2\alpha)}$	$[0, 1/2)$	$O(2^{O(1/(\alpha\varepsilon)^4)} nd \log \frac{k}{\delta})$

Table 1: A comparison of our Sample-and-Search algorithm with state-of-the-art methods. Here, the terms $\epsilon > 0$ and $\delta \in (0, 1)$ are the parameters that control the approximation precision and success probability. Δ denotes the aspect ratio of the given point set.

estimates locally optimal centers dimension-wise across predicted clusters, achieving a $(1 + O(\alpha))$-approximation in $O(nd \log n)$ time when $\alpha \in [0, 1/2)$. Their k-median algorithm significantly improves the approximation guarantee by employing multiple random samplings and pruning techniques.

More recently, Huang et al. (2025) extended the dimension-wise estimation method for learning-augmented k-means to reduce the time complexity while maintaining a similar approximation ratio by using sampling to avoid sorting. They also proposed a k-median algorithm that achieves a $1 + \frac{(6+\epsilon)\alpha - 4\alpha^2}{(1-\alpha)(1-2\alpha)}$-approximation, which represents the state-of-the-art in terms of approximation ratio for learning-augmented k-median clustering, as far as we are aware. However, in their work, the structural differences between k-means and k-median lead to a fundamental algorithmic gap. For k-means, the mean center has a closed-form solution that can be computed independently across dimensions. In contrast, k-median centers lack closed-form expressions and cannot be decomposed dimension-wise, making them significantly harder to compute even when point label predictions are available. As a result, their k-median method needs brute-force grid partitioning and searching procedure in the original high-dimentional space, thus introduces an exponential dependence on d, which is generally considered unacceptable in practice, particularly in high dimensional scenarios. Hence, a key open problem is: *Is it possible to design an algorithm that achieves the state-of-the-art approximation ratio while overcoming the exponential dependence on the dimension d?*

Our key ideas and main contributions. Our key insight is that for each predicted cluster, the true median of the correctly labeled subset lies close to a low-dimensional subspace spanned by a small random sample. This allows us to efficiently discretize the search space using a low-dimensional grid, thus reducing the computational cost. The main technical challenge is that the predicted clusters may be noisy—some points are misclassified, and the predicted cluster center may be far away from the true one. To tackle this obstacle, we design a novel sampling-and-search framework that can effeictively select appropriate candidate cluster centers. The key idea is to utilize a greedy search strategy in the aforementioned low-dimensional grid, which neatly avoids to explicitly distinguish between the correclty labeled and mislabeled points.

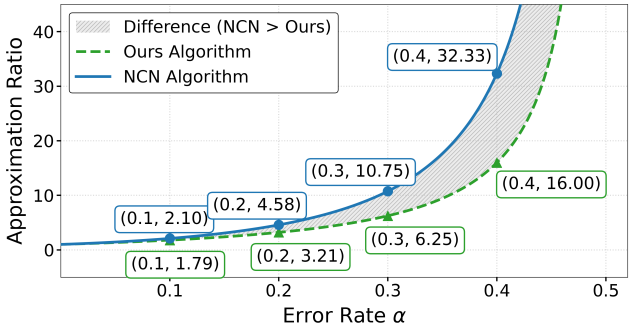


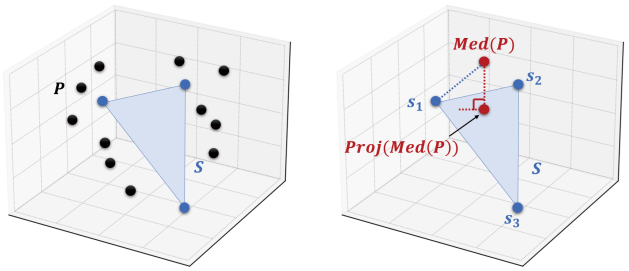
Figure 1: Comparison of the approximation ratios for our algorithm (with $\epsilon = 0.1$) and the NCN algorithm in terms of the error rate α. The plot demonstrates that our algorithm (green dashed line) consistently achieves a lower approximation ratio than the NCN algorithm (Nguyen, Chaturvedi, and Nguyen 2023) (blue solid line) across all values of $\alpha \in [0, 1/2)$. The **grey hatched area** highlights this performance gap, which becomes more pronounced as α increases.

Our contributions are summarized as follows: We propose a simple yet effective algorithm for learning-augmented k-median clustering. The time complexity is **linear in n and** d, avoiding exponential dependence on the dimension d. At the same time, our algorithm achieves an approximation ratio of $1 + \frac{(6+\epsilon)\alpha - 4\alpha^2}{(1-\alpha)(1-2\alpha)}$ for $\alpha < \frac{1}{2}$, matching the state-of-the-art. Furthermore, we conduct a set of experiments on high-dimensional datasets, demonstrating speedups (up to $10\times$) over prior methods while maintaining relatively high clustering quality. Figure 1 compares the approximation ratio of their algorithm (denoted as NCN) with ours. Table 1 provides a detailed comparison for the results of our and existing methods.

1.1 Preliminaries

Notations. Let X denote the input set of n points in $\mathbb{R}^d$. For any two points $p, q \in \mathbb{R}^d$, their Euclidean distance is $\|p - q\|_2$.

Given any point set C, the distance from a point p to its closest point in C is denoted as $\texttt{dist}(p, C) = \min_{c \in C} \|p - c\|_2$. In particular, when C is the given set of centers for the



(a) (b)

Figure 2: (a) provides a simplified illustration of how a subspace is generated. We sample a subset S (denoted by the blue points) from the original point set P (denoted by the black points), and S forms a subspace $\texttt{span}(S)$. (b) shows that $\texttt{span}(S)$ contains a projection of $\texttt{Med}(P)$, denoted by $\texttt{Proj}(\texttt{Med}(P))$, which is close to $\texttt{Med}(P)$. Moreover, S contains a point (*e.g.,* s_1) that is within a bounded distance from $\texttt{Med}(P)$).

point set X, the corresponding cost, denoted as $\texttt{Cost}(X, C)$, is defined as $\texttt{Cost}(X, C) = \sum_{x \in X} \texttt{dist}(p, C)$. With a slight abuse of notation, we also use $\texttt{Cost}(P, c)$ to denote the sum of distances from a point set P to a single point c.

We denote the optimal k clusters for the given instance X as $\{X_1^*, \ldots, X_k^*\}$, and the set $C^* = \{c_1^*, \ldots, c_k^*\}$ contains their corresponding optimal centers.

For any point set P and we use $\texttt{Med}(P)$ to denote its **median point**, i.e.,

$$\texttt{Med}(P) = \arg\min_{q \in \mathbb{R}^d} \sum_{p \in P} \|p - q\|_2. \tag{1}$$

Therefore, for each $1 \leq i \leq k$, $c_i^* = \texttt{Med}(X_i^*)$.

Definition 1.1 (learning-augmented k-median clustering). *Suppose there exists a predictor that outputs a labeled partition $\{\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_k\}$ for X, parameterized by a label error rate $\alpha \in [0, 1)$, which satisfies:*

$$|\tilde{X}_i \cap X_i^*| \geqslant (1 - \alpha) \max(|\tilde{X}_i|, |X_i^*|)$$

where $|\cdot|$ denotes the number of points in a set. The goal of learning-augmented k-median clustering is using such a partially correct result to compute a center set $C \subset \mathbb{R}^d$ that minimizes $\texttt{Cost}(P, C)$.

We also introduce two important propositions on geometric median point in Euclidean space, which are essential for our following proofs.

Proposition 1.1. *(Badoiu, Har-Peled, and Indyk 2002) Let P be a point set in $\mathbb{R}^d$. Given two parameters $1 > \varepsilon > 0$ and $\gamma > 1$, we draw a random sample S from P of size $\frac{\gamma}{\varepsilon^3} \log \frac{1}{\varepsilon}$. Then, with the probability at least $1 - 1/\gamma$, the following two events occur: **(i)** The flat $\texttt{span}(S)$ contains a point within a distance of $\frac{\epsilon \cdot \texttt{Cost}(P, \texttt{Med}(P))}{|P|}$ from $\texttt{Med}(P)$, where $\texttt{span}(S)$ denotes the subspace spanned by the points in S. **(ii)** The set S contains a point within a distance of $2 \times \frac{\texttt{Cost}(P, \texttt{Med}(P))}{|P|}$ from $\texttt{Med}(P)$.*

Proposition 1.1 shows that the subspace spanned by a sufficient random sample from P is guaranteed to contain a good approximation of the $\texttt{Med}(P)$, which enables us to construct a candidate set of centers by partitioning this subspace with a grid to approximate the optimal center. In Figure 2, we depicts the generation of the subspace from the samples. Proposition 1.2 is used to estimate the average cost of the clusters, which in turn guides the design of the grid cell side-length.

Proposition 1.2. *(Kumar, Sabharwal, and Sen 2010) Let P be a point set in $\mathbb{R}^d$. Given a parameter $\zeta \in (0, 1/12)$, we randomly sample a point p_0 and a set S of size $1/\zeta$ from P. Define the value $v = \texttt{Cost}(S, p_0)$. Then with the probability $> \frac{(1-\zeta^2)^{1/\zeta+1}}{2}$, we have:*

$$\frac{v\zeta^3}{2} \leqslant \frac{\texttt{Cost}(P, \text{Med}(P))}{|P|} \leqslant \frac{v}{\zeta}.$$

Upper bound on error rate α. We consider the label error rate α to be upper-bounded by $1/2$. As discussed by Nguyen, Chaturvedi, and Nguyen (2023), when α reaches $1/2$, the relationship between the predicted and optimal clusters can break down entirely.

1.2 Other related work

k-median algorithms. Due to the NP-Hardness of the k-median problem (Cohen-Addad and C.S.Karthik 2019), its approximate algorithms have been extensively studied over the past half-century. Although several PTAS (Polynomial-Time Approximation Scheme) algorithms have been proposed, their running time is exponential in either the dimension d or the number of clusters k (Arora, Raghavan, and Rao 1998; Cohen-Addad, Feldmann, and Saulpic 2021; Kumar, Sabharwal, and Sen 2010), making them impractical for many settings. Other algorithms include the $(3 + \epsilon)$-approximation algorithm via local search proposed by Arya et al. (2001) and the 3.25-approximation using the LP-rounding approach proposed by Charikar and Li (2012).

k-means alogrithms. Similarly with the k-median problem, existing PTAS algorithms for the k-means problem exhibit an exponential dependence on either d or k (Cohen-Addad, Feldmann, and Saulpic 2021; Kumar, Sabharwal, and Sen 2010). Simultaneously, the widely used k-means++ algorithm has a time complexity of $O(ndk)$ and achieves an approximation ratio of $O(\log k)$ (Arthur and Vassilvitskii 2007), or $O(1)$ for well-separated data (Ostrovsky et al. 2013).

2 Our Algorithm And Theoretical Analysis

In this section, we propose the "sample-and-search" algorithm for learning-augmented k-median clustering. Our main idea is to extract information from predicted labels through uniform sampling, then leverage the properties of median point (based on Proposition 1.1) to construct a candidate center set in a low-dimensional subspace, and finally employ a greedy search approach to find the desired solution from the candidate center set. This sample-and-search strategy avoids searching in the original space which may be

much larger than the subspace derived from Proposition 1.1, and thereby is able to reduce the total computational complexity to a great extent. In Section 2.1, we introduce the detailed algorithm and our main theoretical result, i.e., Theorem 2.1. Then, we provide the proof for Theorem 2.1 in Section 2.2.

2.1 Our Proposed Algorithm And Main Theorem

We present the Sample-and-Search algorithm in Algorithm 1. In general, the algorithm consists of three main stages:

1. **Sampling-Based Subspace Construction:** For each predicted cluster, we sample a small subset of points to form a **"basis"** that captures a "neighbor" within a bounded distance from the optimal cluster center. Note that once we have found such a basis, we only need to search within the low-dimensional subspace spanned by this small basis to approximate the optimal center of the cluster. This allows the size of our search space to depend only on ϵ, not on the dimension d.
2. **Grid-based Candidate Generation:** After generating k appropriate subspaces where each one of them is sufficiently close to the corresponding optimal centers, we construct a grid structure in each of the subspaces to generate k small candidate sets of center points. This eliminates the need to search the original high-dimensional space.
3. **Greedy Center Selection:** We select the best center from the candidate set using a cost-minimization greedy selection procedure.

We present the main theoretical result of our algorithm below.

Theorem 2.1. *Let $1 > \epsilon > 0$ and $1 > \delta > 0$ be two parameters. Given an instance X as described in Definition 1.1, if we assume the error rate $\alpha < \frac{1}{2}$, then Algorithm 1 can output a solution with the approximation ratio $\left(1 + \frac{6\alpha - 4\alpha^2 + \epsilon\alpha}{(1-\alpha)(1-2\alpha)}\right)$ with probability $1 - \delta$. The time complexity is $O(2^{O(1/(\alpha\varepsilon)^4)} nd \log \frac{k}{\delta})$.*

2.2 Proof of Theorem 2.1

We divide the proof into three main steps: First, we establish that for each predicted cluster, with high probability, the constructed candidate center set contains at least one point that is close to the true median of the correctly labeled subset of the predicted cluster. This is formalized in Lemma 2.2, which leverages the geometric properties from Proposition 1.1 and Proposition 1.2 under our sampling design. Second, we analyze the cost of the selected center from the candidate set. In Lemma 2.3, we show that this center yields a clustering cost close to the optimal one, despite the noisy labels, by carefully bounding the additional cost incurred by misclassified points and the optimality of the greedy choice.

Finally, we aggregate the bounds over all clusters to obtain the total clustering cost, and analyze the size of the candidate set and runtime of our algorithm.

Algorithm 1: SAMPLE-AND-SEARCH FOR LEARNING-AUGMENTED k-MEDIAN

Input: A k-median instance, consisting of a finite set of points $X \subset \mathbb{R}^d$ and an integer k; a predicted partition $\{\tilde{X}_1, \ldots, \tilde{X}_k\}$ of X with error rate $\alpha \in [0, 1/2)$; an accuracy parameter $\epsilon \in (0, 1)$ and failure probability $\delta \in (0, 1)$.
Output: A set $\hat{C} \in \mathbb{R}^d$ of centers with $|\hat{C}| = k$
Initialize $\hat{C} \leftarrow \emptyset$, $\zeta \leftarrow \frac{1}{13}$
for $i \leftarrow 1$ **to** k **do**
  Initialize a candidate set $C_i \leftarrow \emptyset$
  for $j \leftarrow 1$ **to** $\left\lceil \frac{\log(\delta/k)}{\log(0.975)} \right\rceil$ **do**
    Samplings: first, randomly sample a point $y_i^j \in \tilde{X}_i$, and then sample two separated sets from $\tilde{X}_i$ uniformly at random: a set $Q_i^j \subseteq \tilde{X}_i$ of size $\lceil \frac{2}{(1-\alpha)\zeta} \rceil$, and a set $R_i^j \subseteq \tilde{X}_i$ of size $\lceil \frac{4\log(1/(\frac{\alpha\epsilon}{2}))}{(1-\alpha)(\frac{\alpha\epsilon}{2})^3} \rceil$
    for each subset $Q \subseteq Q_i^j$ of size $1/\zeta$ **do**
      $v \leftarrow \texttt{Cost}(Q, y_i^j)$
      $a \leftarrow \frac{v\zeta^3}{2}$ and $b \leftarrow \frac{v}{\zeta}$
      for each integer $l \in \{\lfloor \log_2 a \rfloor, \ldots, \lceil \log_2 b \rceil\}$ **do**
        $t \leftarrow 2^l$
        Run Algorithm 2: $S \leftarrow \text{CSC}(R_i^j, t, \alpha, \epsilon)$
        $C_i \leftarrow C_i \cup S$
      end for
    end for
  end for
  For each $c \in C_i$, define $N_i(c)$ as the set of $\lceil (1-\alpha)|\tilde{X}_i| \rceil$ points in $\tilde{X}_i$ closest to c
  Greedy selection: find the best candidate for the i-th cluster center, $\hat{c}_i \leftarrow \arg\min_{c \in C_i} \sum_{x \in N_i(c)} \|x - c\|_2$
  $\hat{C} \leftarrow \hat{C} \cup \{\hat{c}_i\}$
end for
return $\hat{C}$

Algorithm 2: CANDIDATE SET CONSTRUCTION (CSC)

Input: A point set $R \subset \mathbb{R}^d$; an approximate average cost $t > 0$; parameters α and ϵ.
Output: A set of candidate centers $S \subset \mathbb{R}^d$.
Initialize $S \leftarrow \emptyset, \theta \leftarrow \frac{\alpha\epsilon t}{4|R|}$
for each point $r \in R$ **do**
  Initialize a candidate set $S_r \leftarrow \emptyset$
  Construct a grid G_r on $\texttt{span}(R)$ centered at r with side-length θ
  Define a ball $B(r, 2t) \leftarrow \{x \in \mathbb{R}^d \mid \|x - r\|_2 \leq 2t\}$
  $S_r \leftarrow G_r \cap B(r, 2t)$
  $S \leftarrow S \cup S_r$
end for
return S

For convenience, we denote the intersection of $\tilde{X}_i$ and X_i^* as T_i, i. e. , $T_i = \tilde{X}_i \cap X_i^*$.

Lemma 2.2. *For predicted cluster $\tilde{X}_i$, with a probability of $1 - \frac{\delta}{k}$, there exists a point $q \in \tilde{C}_i$ satisfying:*

$$||q - \texttt{Med}(T_i)||_2 \leqslant \frac{\alpha\epsilon \times \texttt{Cost}(T_i, \texttt{Med}(T_i))}{|T_i|}. \quad (2)$$

Proof. First, under the learning-augmented setting, we have $|T_i| \geqslant (1-\alpha)\max(|\tilde{X}_i|, |X_i^*|) > \frac{1}{2}|\tilde{X}_i|$. As we uniformly sample a point y_i from $\tilde{X}_i$, and uniformly sample a set Q_i^j from $\tilde{X}_i$ with size $\frac{2}{(1-\alpha)\zeta}$ in the first stage of our algorithm, by employing Markov's inequality, we deduce that, with probability at least $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$, the following two events occur simultaneously:

$$y_i \in T_i, |T_i \cap Q_i^j| \geqslant \frac{1}{\zeta}.$$

Now assume both of these events occur. There exists a subset $Q \subseteq (T_i \cap Q_i^j)$ of size $\frac{1}{\zeta}$. According to Proposition 1.2, if we set $p_0 = y_i^j, S = Q, P = T_i$, for $v = \texttt{Cost}(Q, y_i^j)$, we have

$$\frac{v\zeta^3}{2} \leqslant \frac{\texttt{Cost}(T_i, \texttt{Med}(T_i))}{|T_i|} \leqslant \frac{v}{\zeta} \quad (3)$$

with probability at least $\frac{(1-\zeta^2)^{1/\zeta+1}}{2}$. In the second stage of our algorithm, we iterate over all subsets of Q_i^j of size $\frac{1}{\zeta}$, therefore, there exists an integer l in the interval $\lfloor \log \frac{v\zeta^3}{2}, \log \frac{v}{\zeta} \rceil$ such that $t = 2^l$ satisfies

$$t/2 \leqslant \frac{\texttt{Cost}(T_i, \texttt{Med}(T_i))}{|T_i|} \leqslant t. \quad (4)$$

Similarly, as we uniformly sample a set R_i^j from $\tilde{X}_i$ with size $\frac{4\log(1/\frac{\alpha\epsilon}{2})}{(1-\alpha)(\frac{\alpha\epsilon}{2})^3}$ in the first stage of our algorithm, by employing Markov's inequality, we have $|T_i \cap R_i^j| \geqslant \frac{2\log(1/\frac{\alpha\epsilon}{2})}{(\frac{\alpha\epsilon}{2})^3}$ with probability at least $1/2$. Thus, according to the Proposition 1.1, with probability at least $1/2$, the following two events happen:

1. The flat $\texttt{span}(R_i^j)$ contains a point at a distance $\leqslant \frac{\alpha\epsilon \texttt{Cost}(T_i, \texttt{Med}(T_i))}{2|T_i|}$ from $\texttt{Med}(T_i)$,
2. R_i^j contains a point at a distance $\leqslant 2 \times \frac{\texttt{Cost}(T_i, \texttt{Med}(T_i))}{|T_i|}$ from the center $\texttt{Med}(T_i)$.

So, the flat $\texttt{span}(R_i^j)$ contains a point o such that

$$||o - \texttt{Med}(T_i)||_2 \leqslant \frac{\alpha\epsilon \times \texttt{Cost}(T_i, \texttt{Med}(T_i))}{2|T_i|}. \quad (5)$$

Therefore, under the construction of the grid in Algorithm 2, there must exist a point $q \in S$ satisfying

$$||q - o||_2 \leqslant \frac{\alpha\varepsilon t}{4} \leqslant \frac{\alpha\varepsilon \times \texttt{Cost}(T_i, \texttt{Med}(T_i))}{2|T_i|}, \quad (6)$$

where the second inequality comes from inequality (4). Combining inequality (5) and inequality (6), by triangle inequality, we have

$$||q - \texttt{Med}(T_i)||_2 \leqslant \frac{\alpha\epsilon \times \texttt{Cost}(T_i, \texttt{Med}(T_i))}{|T_i|}.$$

Now, we calculate the probability that all events succeed in a single trial. The combined success probability is $\frac{(1-\zeta^2)^{1/\zeta+1}}{32}$. We have $\frac{(1-\zeta^2)^{1/\zeta+1}}{32} > \frac{e^{-\frac{\zeta}{1-\zeta}}}{32} \geq 0.025$ when $\zeta \in (0, 1/12)$. Here, the first inequality is a direct application of $\ln(1-x) > -\frac{x}{1-x}$ and the second inequality is obtained by leveraging the fact that the function is monotonically decreasing. Therefore, the probability of failure for each trial is less than 0.975. Since we perform $\left\lceil \frac{\log(\delta/k)}{\log(0.975)} \right\rceil$ runs, the overall success probability is therefore greater than $1 - 0.975^{\frac{\log(\delta/k)}{\log(0.975)}} = 1 - \frac{\delta}{k}$. □

We now turn to evaluate the clustering cost incurred by the selected centers. Lemma 2.3 plays a central role in this analysis—it quantifies how far the selected center might be from the true cluster center due to noisy labels and sampling variance, and how this error translates into overall clustering cost.

Lemma 2.3. *For each predicted cluster $\tilde{X}_i$, we have:*

$$\texttt{Cost}(X_i^*, \hat{c}_i) \leqslant \left(1 + \frac{6\alpha - 4\alpha^2 + \alpha\epsilon}{(1-\alpha)(1-2\alpha)}\right) \texttt{Cost}(X_i^*, c_i^*).$$

Proof. A critical component of the analysis is to relate the selected center $\hat{c}_i$to the true center c_i^* and the median of correctly labeled points $\texttt{Med}(T_i)$. We begin by splitting the cost into two parts,

$$\texttt{Cost}(X_i^*, \hat{c}_i) = \texttt{Cost}(T_i, \hat{c}_i) + \texttt{Cost}(X_i^* \backslash T_i, \hat{c}_i) \quad (7)$$

We focus on the term "$\texttt{Cost}(T_i, \hat{c}_i)$" first. To establish the equality, we first compute the additional cost incurred by assigning points in T_i to $\hat{c}_i$ by decomposing the sets T_i and $N_i(\hat{c}_i)$ into three disjoint partitions: $S_1 = T_i \cap N_i(\hat{c}_i), S_2 = T_i \setminus N_i(\hat{c}_i), S_3 = N_i(\hat{c}_i) \setminus T_i$. This implies $T_i = S_1 \cup S_2$ and $N_i(\hat{c}_i) = S_1 \cup S_3$. Then the $\texttt{Cost}(T_i, \hat{c}_i) - \texttt{Cost}(T_i, \texttt{Med}(T_i))$ can be written as

$$\begin{aligned}
&\texttt{Cost}(T_i, \hat{c}_i) - \texttt{Cost}(T_i, \texttt{Med}(T_i)) \\
&= [\texttt{Cost}(S_1, \hat{c}_i) + \texttt{Cost}(S_2, \hat{c}_i)] \\
&\quad - [\texttt{Cost}(S_1, \texttt{Med}(T_i)) + \texttt{Cost}(S_2, \texttt{Med}(T_i))] \\
&= [\texttt{Cost}(S_1, \hat{c}_i) + \texttt{Cost}(S_3, \hat{c}_i)] \\
&\quad - [\texttt{Cost}(S_1, \texttt{Med}(T_i)) + \texttt{Cost}(S_3, \texttt{Med}(T_i))] \\
&\quad - [\texttt{Cost}(S_3, \hat{c}_i) - \texttt{Cost}(S_3, \texttt{Med}(T_i))] \\
&\quad + [\texttt{Cost}(S_2, \hat{c}_i) - \texttt{Cost}(S_2, \texttt{Med}(T_i))] \\
&= [\texttt{Cost}(N_i(\hat{c}_i), \hat{c}_i) - \texttt{Cost}(N_i(\hat{c}_i), \texttt{Med}(T_i))] \\
&\quad - [\texttt{Cost}(N_i(\hat{c}_i) \setminus T_i, \hat{c}_i) - \texttt{Cost}(N_i(\hat{c}_i) \setminus T_i, \texttt{Med}(T_i))] \\
&\quad + [\texttt{Cost}(T_i \setminus N_i(\hat{c}_i), \hat{c}_i) - \texttt{Cost}(T_i \setminus N_i(\hat{c}_i), \texttt{Med}(T_i))]
\end{aligned} \quad (8)$$

We also have $|T_i \setminus N_i(\hat{c}_i))| \leqslant \alpha|\tilde{X}_i|$ and $|N_i(\hat{c}_i)) \setminus T_i| \leqslant \alpha|\tilde{X}_i|$. So we can find an upper bound for $\texttt{Cost}(T_i \setminus$

$N_i(\hat{c}_i), \hat{c}_i) - \texttt{Cost}(T_i \setminus N_i(\hat{c}_i), \texttt{Med}(T_i))$ by triangle inequality as

$$\begin{aligned}&\texttt{Cost}(T_i \setminus N_i(\hat{c}_i), \hat{c}_i) - \texttt{Cost}(T_i \setminus N_i(\hat{c}_i), \texttt{Med}(T_i)) \\ &\leqslant |T_i \setminus N_i| \times ||\texttt{Med}(T_i) - \hat{c}_i||_2 \leqslant \alpha|\tilde{X}_i|||\texttt{Med}(T_i) - \hat{c}_i||_2. \end{aligned} \tag{9}$$

We can obtain inequality $\|\texttt{Med}(T_i) - \hat{c}_i||_2 \leqslant \frac{(2+\alpha\epsilon)\texttt{Cost}(X_i^*, c_i^*)}{(1-2\alpha)|\tilde{X}_i|}$ through triangle inequality (the detailed derivation is provided in the extended version of the paper.). Then we have inequality (9) $\leqslant \frac{(2\alpha+\alpha^2\epsilon)\texttt{Cost}(X_i^*, c_i^*)}{1-2\alpha}$.

Similarly, we obtain

$$\begin{aligned}&\texttt{Cost}(N_i(\hat{c}_i) \setminus T_i, \hat{c}_i) - \texttt{Cost}(N_i(\hat{c}_i) \setminus T_i, \texttt{Med}(T_i)) \\ &\leqslant \frac{(2\alpha + \alpha^2\epsilon)\texttt{Cost}(X_i^*, c_i^*)}{1-2\alpha}. \end{aligned} \tag{10}$$

Now we find a upper bound for $\texttt{Cost}(N_i(\hat{c}_i), \hat{c}_i) - \texttt{Cost}(N_i(\hat{c}_i), \texttt{Med}(T_i))$. Because of our greedy selection, we have $\texttt{Cost}(N_i(\hat{c}_i), \hat{c}_i) \leqslant \texttt{Cost}(N_i(q), q) \leqslant \texttt{Cost}(N_i(\hat{c}_i), q)$. So

$$\begin{aligned}&\texttt{Cost}(N_i(\hat{c}_i), \hat{c}_i) - \texttt{Cost}(N_i(\hat{c}_i), \texttt{Med}(T_i)) \\ &\leqslant \texttt{Cost}(N_i(\hat{c}_i), q) - \texttt{Cost}(N_i(\hat{c}_i), \texttt{Med}(T_i)) \\ &\leqslant |(1-\alpha)|\tilde{X}_i|| \times ||q - \texttt{Med}(T_i)||_2. \end{aligned} \tag{11}$$

By applying Lemma 2.2, we can obtain inequality (11) $\leqslant \alpha\epsilon \times \texttt{Cost}(T_i, \texttt{Med}(T_i))$. Putting inequality (9), inequality (10) and inequality (11) together, we obtain the following bound for the left side of equation (7)

$$\begin{aligned}&\texttt{Cost}(T_i, \hat{c}_i) - \texttt{Cost}(T_i, c_i^*) \\ &\leqslant \texttt{Cost}(T_i, \hat{c}_i) - \texttt{Cost}(T_i, \texttt{Med}(T_i)) \\ &\leqslant \frac{(4\alpha + \epsilon\alpha)\texttt{Cost}(X_i^*, c_i^*)}{1-2\alpha}. \end{aligned} \tag{12}$$

Next, we consider the second term "$\texttt{Cost}(X_i^* \setminus T_i, \hat{c}_i)$" in equation (7). By triangle inequality

$$\begin{aligned}\texttt{Cost}(X_i^* \setminus T_i, \hat{c}_i) &\leqslant \texttt{Cost}(X_i^* / T_i, c_i^*) \\ &\quad + |X_i^* \setminus T_i| \times ||\hat{c}_i - c_i^*||_2, \end{aligned} \tag{13}$$

Subsequently, we bound $|X_i^* \setminus T_i|$. It follows from Definition 1.1 that $(1-\alpha)|X_i^*| \leqslant |T_i|, |T_i| \leqslant |\tilde{X}_i|$, so, we can bound $|X_i^* \setminus T_i|$ as

$$|X_i^* \setminus T_i| = |X_i^*| - |T_i| \leqslant \alpha|X_i^*| \leqslant \frac{\alpha|\tilde{X}_i|}{1-\alpha}. \tag{14}$$

Combining (13), (14), and the inequality $\|\hat{c}_i - c_i^*||_2 \leqslant \frac{(2+\alpha\epsilon)\texttt{Cost}(X_i^*, c_i^*)}{(1-2\alpha)|\tilde{X}_i|}$ we obtain in the extended version of the paper, we have

$$\begin{aligned}&\texttt{Cost}(X_i^* \setminus T_i, \hat{c}_i) - \texttt{Cost}(X_i^* \setminus T_i, c_i^*) \\ &\leqslant \frac{\alpha(2 + \alpha\epsilon)\texttt{Cost}(X_i^*, c_i^*)}{(1-\alpha)(1-2\alpha)}. \end{aligned} \tag{15}$$

Now, we derive the final approximation guarantee. Combining inequality (12) and inequality (15), we have

$$\begin{aligned}\texttt{Cost}(X_i^*, \hat{c}_i) &= \texttt{Cost}(T_i, \hat{c}_i) + \texttt{Cost}(X_i^* / T_i, \hat{c}_i) \\ &\leqslant \left(1 + \frac{6\alpha - 4\alpha^2 + \epsilon\alpha}{(1-\alpha)(1-2\alpha)}\right) \texttt{Cost}(X_i^*, c_i^*). \end{aligned}$$

□

We now proceed to formally prove Theorem 2.1 by establishing both the approximation ratio and the runtime complexity.

Proof of Theorem 2.1. Our first step is to compute the approximation ratio of the algorithm. In each cluster, by Lemma 2.3, we obtain

$$\texttt{Cost}(X_i^*, \hat{c}_i) \leqslant \left(1 + \frac{6\alpha - 4\alpha^2 + \epsilon\alpha}{(1-\alpha)(1-2\alpha)}\right) \texttt{Cost}(X_i^*, c_i^*).$$

Therefore, for the entire instance, we have

$$\begin{aligned}&\sum_{i \in [k]} \texttt{Cost}(X_i^*, \{\widehat{c}_j\}_{j=1}^k) \\ &\leqslant \left(1 + \frac{6\alpha - 4\alpha^2 + \epsilon\alpha}{(1-\alpha)(1-2\alpha)}\right) \sum_{i \in [k]} \texttt{Cost}(X_i^*, c_i^*). \end{aligned}$$

We now assess the time complexity of the algorithm. This involves analyzing the size of the candidate center set generated via sampling and grid discretization, and the cost incurred in evaluating all candidate centers.

First, we compute the size of set of candidate centers. The size of the candidate center set we ultimately construct is

$$\begin{aligned}&O\left(|R| \left(\frac{1}{(\alpha\epsilon)^4} \log \frac{1}{(\alpha\epsilon)}\right)^{O(|R|)} \log \frac{k}{\delta}\right) \\ &= O\left(2^{O(\frac{1}{(\alpha\epsilon)^3} \log^2 \frac{1}{(\alpha\epsilon)})} \log \frac{k}{\delta}\right) \leqslant O\left(2^{O(\frac{1}{(\alpha\epsilon)^4})} \log \frac{k}{\delta}\right). \end{aligned}$$

For each candidate point within the candidate center set, the time needed to calculate its cost is $|\tilde{X}_i|d$. Consequently, the overall time complexity of the algorithm is

$$\sum_{i \in [k]} 2^{O(\frac{1}{(\alpha\epsilon)^4})} |\tilde{X}_i| d \log \frac{k}{\delta} = 2^{O(\frac{1}{(\alpha\epsilon)^4})} nd \log \frac{k}{\delta}.$$

Next, we analyze the success probability of the algorithm. As we obtained in Lemma 2.2, the success probability in each cluster of the algorithm is $1 - \frac{\delta}{k}$, therefore, by the union bound, the overall success probability of the algorithm $\geqslant 1 - k \times \frac{\delta}{k} = 1 - \delta$. □

3 Experiment

We evaluated our algorithms on real-world datasets. The experiments were conducted on a server with an Intel(R) Xeon(R) Gold 6154 CPU and 1024GB of RAM. For all experiments, we report the average clustering cost and its standard deviation over 10 independent runs.

Datasets. Following the work of Nguyen, Chaturvedi, and Nguyen (2023), Ergun et al. (2022) and Huang et al. (2025), we evaluate our algorithms on the CIFAR-10 ($n = 50,000, d = 3,072$) (Krizhevsky and Hinton 2009), PHY ($n = 10,000, d = 50$) (KDD 2004), and MNIST ($n = 1,797, d = 64$) (Deng 2012) datasets using a range of error rates α. We additionally evaluated our algorithm's performance on another high dimensional dataset Fashion-MNIST ($n = 60000, d = 784$) (Xiao, Rasul, and Vollgraf 2017)

Predictor Generation and Error Simulation To evaluate our algorithms, we first computed a ground-truth partition for each dataset using Lloyd's algorithm initialized with KMedoids++(denoteed as KMed++). We then generated corrupted partitions with the error rate, α, by randomly selecting an α fraction of points in each true cluster and reassigning them to randomly chosen cluster(denoted as Predictor). To ensure a fair comparison, every algorithm was tested on the exact same set of corrupted labels for any given error rate α.

Algorithms In our experiments, we evaluate our proposed Sample-and-Search algorithm. We compare its performance against other state-of-the-art learning-augmented methods, including the algorithm from Ergun et al. (2022) (denoted as EFS+), Nguyen, Chaturvedi, and Nguyen (2023) (denoted as NCN) and the recent work by Huang et al. (2025) (denoted as HFH+). As noted by Nguyen, Chaturvedi, and Nguyen (2023), the true error rate α is generally unknown in practice, which necessitates a search for its optimal value. To ensure a fair comparison, we implement a uniform hyperparameter tuning strategy for all evaluated algorithms. Specifically, we iterate over 10 candidate values for α, which are chosen from uniformly spaced points in the interval $[0.01, 0.5]$. For each method, the candidate value that minimizes the resulting k-median clustering cost is chosen to produce the final output. To assess the final clustering quality against the ground-truth labels, we additionally report the *Adjusted Rand Index (ARI)* and *Normalized Mutual Information (NMI)* in the extended version of the paper.

Results We present a comparative evaluation of our algorithm against those baselines in Table 2 and Table 3. Table 2 details the performance on the Fashion-MNIST ($n = 60000, d = 784$) for a fixed $k = 10$ across a range of α values. Table 3 shows the results on the PHY ($n = 10,000, d = 50$) with a fixed $\alpha = 0.2$ for various choices of k. Both sets of results demonstrate that our algorithm is substantially faster than all competing methods while generally achieving better approximation quality. Additional experiments on other datasets and a more detailed presentation of the results are available in the extended version. On these datasets, our algorithm also demonstrates significant advantages in terms of both running time and cost.

4 Conclusion and Future work

In this paper, we study the learning-augmented k-median clustering problem. We first introduce an algorithm for this problem based on a simple yet effective sampling method, then study its quality guarantees in theory, and finally conduct a set of experiments to compare with other learning-augmented k-median algorithms. Both theoretical and experimental results demonstrate that our method achieves the state-of-the-art approximation ratio with higher efficiency than existing methods. Following this work, there are several opportunities to further improve our methods from both theoretical and practical perspectives. For example, is it possible to further reduce the time complexity of the algorithm by mitigating or eliminating the exponential dependence on ϵ? Can the approximation ratio be further improved without a significant increase in time complexity? Furthermore, could a learning-augmented clustering algorithm be designed for the streaming model to more effectively handle large-scale data?

Condition		Cost		Time(s)	
α	Method	Avg.	Std. Dev.	Avg.	Std. Dev.
0	KMed++	8.4054e+07	-	-	-
0.1	Predictor	8.4259e+07	-	-	-
	EFS+	8.4050e+07	115.17	270.47	12.97
	HFH+	8.4049e+07	834.92	749.72	18.47
	NCN	8.4050e+07	181.80	272.22	4.37
	Ours	**8.4048e+07**	933.64	**47.37**	0.78
0.2	Predictor	8.4935e+07	-	-	-
	EFS+	8.4057e+07	287.83	283.13	24.07
	HFH+	8.4053e+07	1598.91	751.66	25.42
	NCN	8.4057e+07	309.52	282.97	13.78
	Ours	**8.4052e+07**	961.03	**47.96**	3.33
0.3	Predictor	8.6223e+07	-	-	-
	EFS+	8.4076e+07	467.75	282.57	8.66
	HFH+	8.4065e+07	3527.00	751.13	22.67
	NCN	8.4077e+07	695.49	299.50	22.54
	Ours	**8.4062e+07**	3848.20	**45.38**	1.33
0.4	Predictor	8.8209e+07	-	-	-
	EFS+	8.4109e+07	631.67	297.27	13.66
	HFH+	8.4101e+07	11512.25	758.16	29.89
	NCN	8.4111e+07	1206.20	302.45	25.71
	Ours	**8.4100e+07**	12684.42	**45.29**	2.18
0.5	Predictor	9.0897e+07	-	-	-
	EFS+	8.4150e+07	1320.47	304.67	11.82
	HFH+	8.4148e+07	12671.01	751.08	26.53
	NCN	8.4152e+07	2503.70	305.95	10.98
	Ours	**8.4145e+07**	17385.62	**47.87**	2.04

Table 2: Performance comparison on Fashion-MNIST dataset with $k = 10$ and varied α.

Condition		Cost		Time(s)	
k	Method	Avg.	Std. Dev.	Avg.	Std. Dev.
10	KMed++	2.0224e+08	-	-	-
	Predictor	2.0427e+08	-	-	-
	EFS+	2.0204e+08	4444.80	362.31	52.30
	HFH+	2.0147e+08	105661.96	42.33	1.91
	NCN	2.0163e+08	109131.95	160.15	14.33
	Ours	**2.0134e+08**	82812.18	**20.72**	0.59
30	KMed++	8.4404e+07	-	-	-
	Predictor	8.5018e+07	-	-	-
	EFS+	8.4490e+07	721.59	294.21	54.13
	HFH+	8.4404e+07	4372.98	42.26	2.05
	NCN	8.4480e+07	14266.80	221.81	49.30
	Ours	**8.4404e+07**	3043.38	**27.08**	3.26
50	KMed++	6.2758e+07	-	-	-
	Predictor	6.3111e+07	-	-	-
	EFS+	6.2796e+07	503.13	285.51	22.48
	HFH+	6.2755e+07	1072.71	44.69	1.34
	NCN	6.2791e+07	5662.71	208.89	29.26
	Ours	**6.275456e+07**	677.03	**36.87**	0.73

Table 3: Performance comparison on PHY dataset with fixed $\alpha = 0.2$ and varied k.

Acknowledgments

This work was supported in part by the NSFC through grants No. 62432016 and No. 62272432, the National Key R&D program of China through grant 2021YFA1000900, and the Provincial NSF of Anhui through grant 2208085MF163. The authors would also like to thank the anonymous reviewers for their valuable comments and suggestions.

References

Antoniadis, A.; Gouleakis, T.; Kleer, P.; and Kolev, P. 2020. Secretary and online matching problems with machine learned advice. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Arora, S.; Raghavan, P.; and Rao, S. 1998. Approximation schemes for Euclidean k-medians and related problems. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, 106–113. New York, NY, USA: Association for Computing Machinery. ISBN 0897919629.

Arthur, D.; and Vassilvitskii, S. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, 1027–1035. USA: Society for Industrial and Applied Mathematics. ISBN 9780898716245.

Arya, V.; Garg, N.; Khandekar, R.; Meyerson, A.; Munagala, K.; and Pandit, V. 2001. Local search heuristic for k-median and facility location problems. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, 21–29. New York, NY, USA: Association for Computing Machinery. ISBN 1581133499.

Badoiu, M.; Har-Peled, S.; and Indyk, P. 2002. Approximate clustering via core-sets. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, 250–257. New York, NY, USA: Association for Computing Machinery. ISBN 1581134959.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by swapping assignments between views. In *Advances in Neural Information Processing Systems*, volume 33, 9912–9924.

Charikar, M.; and Li, S. 2012. A dependent LP-rounding approach for the k-median problem. In *Proceedings of the 39th International Colloquium Conference on Automata, Languages, and Programming - Volume Part I*, ICALP'12, 194–205. Berlin, Heidelberg: Springer-Verlag. ISBN 9783642315930.

Cohen-Addad, V.; and C.S.Karthik. 2019. Inapproximability of Clustering in Lp Metrics. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, 519–539.

Cohen-Addad, V.; Feldmann, A. E.; and Saulpic, D. 2021. Near-linear Time Approximation Schemes for Clustering in Doubling Metrics. *J. ACM*, 68(6).

Deng, L. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine*, 29(6): 141–142.

Dinitz, M.; Im, S.; Lavastida, T.; Moseley, B.; and Vassilvitskii, S. 2021. Faster matchings via learned duals. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713845393.

Ergun, J. C.; Feng, Z.; Silwal, S.; Woodruff, D.; and Zhou, S. 2022. Learning-Augmented k-means Clustering. In *International Conference on Learning Representations*.

Gamlath, B.; Lattanzi, S.; Norouzi-Fard, A.; and Svensson, O. 2022. Approximate Cluster Recovery from Noisy Labels. In Loh, P.-L.; and Raginsky, M., eds., *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, 1463–1509. PMLR.

Ghaffari, M.; Mosavi, A.; and Shamshirband, S. 2021. Clustering and high-dimensional representation of social network users' behavior for bot detection. In *Companion Proceedings of the Web Conference 2021*, 19–22.

Hsu, C.-Y.; Indyk, P.; Katabi, D.; and Vakilian, A. 2018. Learning-Based Frequency Estimation Algorithms. In *International Conference on Learning Representations*.

Huang, J.; Feng, Q.; Huang, Z.; Zhang, Z.; Xu, J.; and Wang, J. 2025. New Algorithms for the Learning-Augmented k-means Problem. In *The Thirteenth International Conference on Learning Representations*.

KDD. 2004. KDD Cup. https://osmot.cs.cornell.edu/kddcup/index.html. Accessed: 2025-5-20.

Kiselev, V. Y.; Kirschner, K.; Schaub, M. T.; Andrews, T.; Yiu, A.; Chandra, T.; Natarajan, K. N.; Reik, W.; Barahona, M.; Green, A. R.; and Hemberg, M. 2017. SC3: consensus clustering of single-cell RNA-seq data. *Nature methods*, 14(5): 483–486.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario.

Kumar, A.; Sabharwal, Y.; and Sen, S. 2010. Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM*, 57(2).

Lykouris, T.; and Vassilvitskii, S. 2021. Competitive Caching with Machine Learned Advice. *J. ACM*, 68(4).

Mitzenmacher, M.; and Vassilvitskii, S. 2022. Algorithms with predictions. *Commun. ACM*, 65(7): 33–35.

Nguyen, T. D.; Chaturvedi, A.; and Nguyen, H. 2023. Improved Learning-augmented Algorithms for k-means and k-medians Clustering. In *The Eleventh International Conference on Learning Representations*.

Ostrovsky, R.; Rabani, Y.; Schulman, L. J.; and Swamy, C. 2013. The effectiveness of lloyd-type methods for the k-means problem. *J. ACM*, 59(6).

Roughgarden, T. 2021. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *ArXiv*, abs/1708.07747.