Coresets for the Nearest-Neighbor Rule

Alejandro Flores-Velazco
Department of Computer Science, University of Maryland, College Park, MD, USA
afloresv@cs.umd.edu

David M. Mount
Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, USA
mount@umd.edu

Abstract

Given a training set P of *labeled* points, the *nearest-neighbor rule* predicts the class of an *unlabeled* query point as the label of its closest point in the set. To improve the time and space complexity of classification, a natural question is how to reduce the training set without significantly affecting the accuracy of the nearest-neighbor rule. *Nearest-neighbor condensation* deals with finding a subset $R \subseteq P$ such that for every point $p \in P$, p's nearest-neighbor in R has the same label as p. This relates to the concept of *coresets*, which can be broadly defined as subsets of the set, such that an exact result on the coreset corresponds to an approximate result on the original set. However, the guarantees of a coreset hold for any query point, and not only for the points of the training set.

This paper introduces the concept of coresets for nearest-neighbor classification. We extend existing criteria used for condensation, and prove sufficient conditions to correctly classify any query point when using these subsets. Additionally, we prove that finding such subsets of minimum cardinality is NP-hard, and propose quadratic-time approximation algorithms with provable upper-bounds on the size of their selected subsets. Moreover, we show how to improve one of these algorithms to have subquadratic runtime, being the first of this kind for condensation.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases coresets, nearest-neighbor rule, classification, nearest-neighbor condensation, training-set reduction, approximate nearest-neighbor, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.ESA.2020.47

Related Version A full version of the paper is available at `https://arxiv.org/abs/2002.06650`.

Supplementary Material Source code is available at `https://github.com/afloresv/nnc`.

Funding Research partially supported by NSF grant CCF-1618866.

Acknowledgements Thanks to Prof. Emely Arráiz for suggesting the problem of condensation while the first author was a student at Universidad Simón Bolívar, Venezuela. Thanks to Ahmed Abdelkader for the helpful discussions and valuable suggestions.

1 Introduction

In non-parametric classification, we are given a *training set* P consisting of n points in a metric space $(\mathcal{X}, \mathsf{d})$, with domain $\mathcal{X}$ and distance function $\mathsf{d} : \mathcal{X}^2 \to \mathbb{R}^+$. Additionally, P is partitioned into a finite set of *classes* by associating each point $p \in P$ with a *label* $l(p)$, indicating the class to which it belongs. Given an *unlabeled* query point $q \in \mathcal{X}$, the goal of a *classifier* is to predict q's label using the training set P.

The *nearest-neighbor rule* is among the best-known classification techniques [19]. It assigns a query point the label of its closest point in P, according to the metric d. The nearest-neighbor rule exhibits good classification accuracy both experimentally and theoretically [14, 15, 36], but it is often criticized due to its high space and time complexities. Clearly, the training set


© Alejandro Flores-Velazco and David M. Mount;
licensed under Creative Commons License CC-BY

P must be stored to answer nearest-neighbor queries, and the time required for such queries depends to a large degree on the size and dimensionality of the data. These drawbacks inspire the question of whether it is possible replace P with a significantly smaller subset, without significantly reducing the classification accuracy under the nearest-neighbor rule. This problem is called *nearest-neighbor condensation* [22, 25, 34, 37].

There are obvious parallels between condensation and the concept of *coresets* in geometric approximation [1, 17, 23, 33]. Intuitively, a *coreset* is small subset of the original data, that well approximates some statistical properties of the original set. Coresets have also been applied to many problems in machine learning, such as clustering and neural network compression [8, 11, 18, 29]. This includes recent results on coresets for the SVM classifier [38].

This paper presents the first approach to compute coresets for the nearest-neighbor rule, leveraging its resemblance to the problem of nearest-neighbor condensation. We also present one of the first results on practical condensation algorithms with theoretical guarantees.

Preliminaries. Given any point $q \in \mathcal{X}$ in the metric space, its nearest-neighbor, denoted $\mathrm{nn}(q)$, is the closest point of P according the the distance function d. The distance from q to its nearest-neighbor is denoted by $\mathsf{d}_{\mathrm{nn}}(q, P)$, or simply $\mathsf{d}_{\mathrm{nn}}(q)$ when P is clear. Given a point $p \in P$ from the training set, its nearest-neighbor in P is point p itself. Additionally, any point of P whose label differs from p's is called an *enemy* of p. The closest such point is called p's *nearest-enemy*, and the distance to this point is called p's *nearest-enemy distance*. These are denoted by $\mathrm{ne}(p)$ and $\mathsf{d}_{\mathrm{ne}}(p, P)$ (or simply $\mathsf{d}_{\mathrm{ne}}(p)$), respectively.

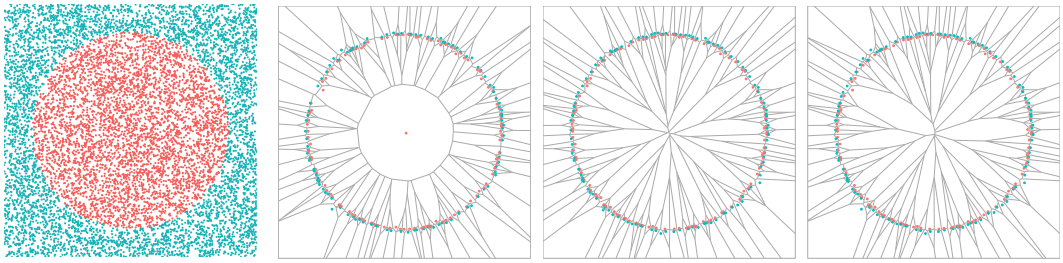
Clearly, the size of a coreset for nearest-neighbor classification depends on the spatial characteristics of the classes in the training set. For example, it is much easier to find a small coreset for two spatially well separated clusters than for two classes that have a high degree of overlap. To model the intrinsic complexity of nearest-neighbor classification, we define κ to be the number of nearest-enemy points of P, i.e., the cardinality of set $\{\mathrm{ne}(p) \mid p \in P\}$.

Through a suitable uniform scaling, we may assume that the *diameter* of P (that is, the maximum distance between any two points in the training set) is 1. The *spread* of P, denoted as Δ, is the ratio between the largest and smallest distances in P. Define the *margin* of P, denoted γ, to be the smallest nearest-enemy distance in P. Clearly, $1/\gamma \leq \Delta$.

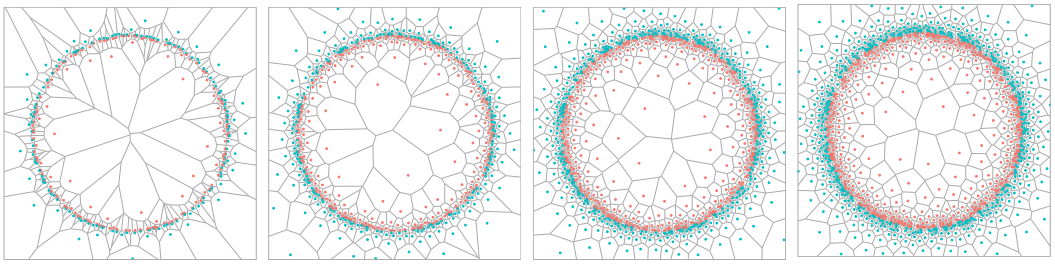
A metric space $(\mathcal{X}, \mathsf{d})$ is said to be *doubling* [26] if there exist some bounded value λ such that any metric ball of radius r can be covered with at most λ metric balls of radius $r/2$. Its *doubling dimension* is the base-2 logarithm of λ, denoted as $\mathrm{ddim}(\mathcal{X}) = \log \lambda$. Throughout, we assume that $\mathrm{ddim}(\mathcal{X})$ is a constant, which means that multiplicative factors depending on $\mathrm{ddim}(\mathcal{X})$ may be hidden in our asymptotic notation. Many natural metric spaces of interest are doubling, including d-dimensional Euclidean space whose doubling dimension is $\Theta(d)$. It is well know that for any subset $R \subseteq \mathcal{X}$ with some spread Δ_R, the size of R is bounded by $|R| \leq \lceil \Delta_R \rceil^{\mathrm{ddim}(\mathcal{X})+1}$.

Related Work. A subset $R \subseteq P$ is said to be *consistent* [25] if and only if for every $p \in P$ its nearest-neighbor in R is of the same class as p. Intuitively, R is consistent if and only if all points of P are correctly classified using the nearest-neighbor rule over R. Formally, the problem of *nearest-neighbor condensation* consists of finding a consistent subset of P.

Another criterion used for condensation is known as *selectiveness* [34]. A subset $R \subseteq P$ is said to be *selective* if and only if for all $p \in P$ its nearest-neighbor in R is closer to p than its nearest-enemy in P. Clearly, any selective subset is also consistent. Observe that these condensation criteria ensure that every point in the training set will be correctly classified after condensation, but they do not imply the same for arbitrary points in the metric space.



(a) Training set (10^4 pts). **(b)** FCNN (222 pts). **(c)** VSS (233 pts). **(d)** RSS (233 pts).



(e) 0.1-RSS (300 pts). **(f)** 0.5-RSS (540 pts). **(g)** 1-RSS (846 pts). **(h)** $\sqrt{2}$-RSS (1066 pts).

Figure 1 An illustrative example of the subsets selected by different condensation algorithms from an initial training set P in $\mathbb{R}^2$ of 10^4 points. FCNN, VSS, and RSS, are known algorithms for this problem, while α-RSS is proposed in this paper, along with new condensation criteria. The subsets selected by α-RSS depend on the parameter $\alpha \geq 0$, here assigned to the values $\alpha = \{0.1, 0.5, 1, \sqrt{2}\}$.

It is known that the problems of computing consistent and selective subsets of minimum cardinality are both NP-hard [28, 39, 40]. An approximation algorithm called NET [22] was proposed for the problem of finding minimum cardinality consistent subsets, along with almost matching hardness lower-bounds. The algorithm simply computes a γ-net of P, where γ is the minimum nearest-enemy distance in P, which clearly results in a consistent subset of P (also selective). In practice, γ tends to be small, which results in subsets of much higher cardinality than needed. To overcome this issue, the authors proposed a post-processing pruning technique to further reduce the selected subset. Even with the extra pruning, NET is often outperformed on typical data sets by more practical heuristics with respect to runtime and selection size. More recently, a subexponential-time algorithm was proposed [10] for finding minimum cardinality consistent subsets of point sets $P \subset \mathbb{R}^2$ in the plane, along with other case-specific algorithms for special instances of the problem in $\mathbb{R}^2$. On the other hand, less is known about computing minimum cardinality selective subsets: there is only a worst-case exponential time algorithm called SNN [34] for computing such optimal subsets.

Most research has focused on proposing practical heuristics to find either consistent or selective subsets of P (for comprehensive surveys see [27, 37]). CNN (*Condensed Nearest-Neighbor*) [25] was the first algorithm proposed to compute consistent subsets. Even though it has been widely used in the literature, CNN suffers from several drawbacks: its running time is cubic in the worst-case, and the resulting subset is *order-dependent*, meaning that the result is determined by the order in which points are considered by the algorithm. Alternatives include FCNN (*Fast* CNN) [3] and MSS (*Modified Selective Subset*) [7], which compute consistent and selective subsets respectively. Both algorithms run in $\mathcal{O}(n^2)$ worst-case time, and are order-independent. While such heuristics have been extensively studied experimentally [21], theoretical results are scarce. Recently, it was shown [20] that the size of the subsets selected by MSS cannot be bounded, while for FCNN it is still unknown whether is possible to establish any bound. The same paper [20] proposes two new algorithms, namely

RSS (*Relaxed Selective Subset*) and VSS (*Voronoi Selective Subset*), to find selective subsets of P in $\mathcal{O}(n^2)$ worst-case time. Both algorithms provide some guarantees on its selection size in Euclidean space.

Contributions. As mentioned in the previous section, consistency and selectivity imply correct classification to points of the training set, but not to arbitrary points of the metric space (This is striking since this is the fundamental purpose of classification!). In this paper, we introduce the concept of a coreset for classification with the nearest-neighbor rule, which provides approximate guarantees on correct classification for all query points. We demonstrate their existence, analyze their size, and discuss their efficient computation.

We say that a subset $R \subseteq P$ is an *ε-coreset for the nearest-neighbor rule* on P, if and only if for every query point $q \in \mathcal{X}$, the class of its exact nearest-neighbor in R is the same as the class of some ε-approximate nearest-neighbor of q in P (see Section 2 for definitions). Recalling the concepts of κ and γ introduced in the preliminaries, here is our main result:

▶ **Theorem 1.** *Given a training set P in a doubling metric space $(\mathcal{X}, \mathsf{d})$, there exist an ε-coreset for the nearest-neighbor rule of size $\mathcal{O}(\kappa \log \frac{1}{\gamma} (1/\varepsilon)^{\mathrm{ddim}(\mathcal{X})+1})$, and this coreset can be computed in subquadratic worst-case time.*

Here is a summary of our principal results:

- We extend the criteria used for nearest-neighbor condensation, and identify sufficient conditions to guarantee the correct classification of any query point after condensation.
- We prove that finding minimum-cardinality subsets with this new criteria is NP-hard.
- We provide quadratic-time approximation algorithms with provable upper-bounds on the sizes of their selected subsets, and we show that the running time of one such algorithm can be improved to be subquadratic.

Our subquadratic-time algorithm is the first with such worst-case runtime for the problem of nearest-neighbor condensation.

2 Coreset Characterization

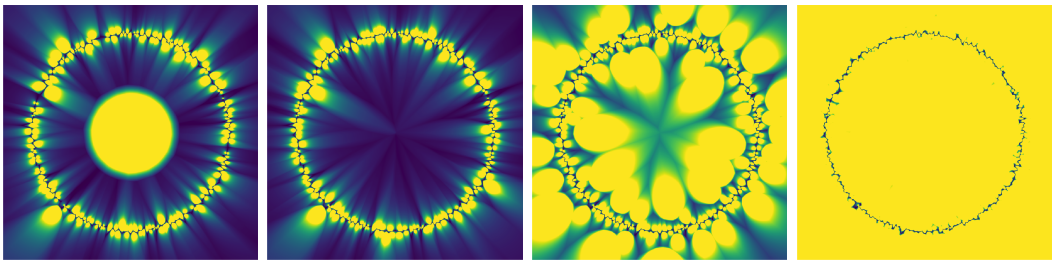
In practice, nearest-neighbors are usually not computed exactly, but rather approximately. Given an approximation parameter $\varepsilon \geq 0$, an *ε-approximate* nearest-neighbor or ε-ANN query returns any point whose distance from the query point is within a factor of $(1 + \varepsilon)$ times the true nearest-neighbor distance.

Intuitively, a query point should be easier to classify if its nearest-neighbor is significantly closer than its nearest-enemy. This intuition can be formalized through the concept of the *chromatic density* [31] of a query point $q \in \mathcal{X}$ with respect to a set $R \subseteq P$, defined as:

$$\delta(q, R) = \frac{\mathsf{d}_{\mathrm{ne}}(q, R)}{\mathsf{d}_{\mathrm{nn}}(q, R)} - 1. \tag{1}$$

Clearly, if $\delta(q, R) > \varepsilon$ then q will be correctly classified[1] by an ε-ANN query over R, as all possible candidates for the approximate nearest-neighbor belong to the same class as q's true nearest-neighbor. However, as evidenced in Figures 2a and 2b, one side effect of existing condensation algorithms is a significant reduction in the chromatic density of query points. Consequently, we propose new criteria and algorithms that maintain high chromatic densities after condensation, which are then leveraged to build coresets for the nearest-neighbor rule.

[1] By *correct classification*, we mean that the classification is the same as the classification that results from applying the nearest-neighbor rule exactly on the entire training set P.



(a) FCNN. **(b)** RSS. **(c)** 0.1-RSS. **(d)** 0.5-RSS.

Figure 2 Heatmap of *chromatic density* values of points in $\mathbb{R}^2$ w.r.t. the subsets computed by different condensation algorithms: FCNN, RSS, and α-RSS (see Figure 1). *Yellow* • corresponds to chromatic density values ≥ 0.5, while *blue* • corresponds to 0. Evidently, α-RSS helps maintaining high chromatic density values when compared to standard condensation algorithms.

2.1 Approximation-Sensitive Condensation

The decision boundaries of the nearest-neighbor rule (that is, points q such that $\mathsf{d}_{\text{ne}}(q, P) = \mathsf{d}_{\text{nn}}(q, P)$) are naturally characterized by points that separate clusters of points of different classes. As illustrated in Figures 1b-1d, condensation algorithms tend to select such points. However, this behavior implies a significant reduction of the chromatic density of query points that are far from such boundaries (see Figures 2a-2b).

A natural way to define an approximate notion of consistency is to ensure that all points in P are correctly classified by ANN queries over the condensed subset R. Given a condensation parameter $\alpha \geq 0$, we define a subset $R \subseteq P$ to be:

α-consistent if $\forall\ p \in P,\ \mathsf{d}_{\text{nn}}(p, R) < \mathsf{d}_{\text{ne}}(p, R)/(1+\alpha)$.

α-selective if $\forall\ p \in P,\ \mathsf{d}_{\text{nn}}(p, R) < \mathsf{d}_{\text{ne}}(p, P)/(1+\alpha)$.

It is easy to see that the standard forms arise as special cases when $\alpha = 0$. These new condensation criteria imply that $\delta(p, R) > \alpha$ for every $p \in P$, meaning that p is correctly classified using an α-ANN query on R. Note that any α-selective subset is also α-consistent. Such subsets always exist for any $\alpha \geq 0$ by taking $R = P$. Current condensation algorithms cannot guarantee either α-consistency or α-selectiveness unless α is equal to zero. Therefore, the central algorithmic challenge is how to efficiently compute such sets whose sizes are significantly smaller than P. We propose new algorithms to compute such subsets, which showcase how to maintain high chromatic density values after condensation, as evidenced in Figures 2c and 2d. This empirical evidence is matched with theoretical guarantees for α-consistent and α-selective subsets, described in the following section.

2.2 Guarantees on Classification Accuracy

These newly defined criteria for nearest-neighbor condensation enforce lower-bounds on the chromatic density of any point of P after condensation. However, this doesn't immediately imply having similar lower-bounds for unlabeled query points of $\mathcal{X}$. In this section, we prove useful bounds on the chromatic density of query points, and characterize sufficient conditions to correctly classify some of these query points after condensation.

Intuitively, the chromatic density determines how easy it is to correctly classify a query point $q \in \mathcal{X}$. We show that the "ease" of classification of q after condensation (i.e., $\delta(q, R)$) depends on both the condensation parameter α, and the chromatic density of q before condensation (i.e., $\delta(q, P)$). This result is formalized in the following lemma:

▶ **Lemma 2.** *Let $q \in \mathcal{X}$ be a query point, and R an α-consistent subset of P, for $\alpha \geq 0$. Then, q's chromatic density with respect to R is:*

$$\delta(q, R) > \frac{\alpha\,\delta(q, P) - 2}{\delta(q, P) + \alpha + 3}.$$

Proof. The proof follows by analyzing q's nearest-enemy distance in R. To this end, consider the point $p \in P$ that is q's nearest-neighbor in P. There are two possible cases:

Case 1: If $p \in R$, clearly $\mathsf{d}_{\mathrm{nn}}(q, R) = \mathsf{d}_{\mathrm{nn}}(q, P)$. Additionally, it is easy to show that after condensation, q's nearest-enemy distance can only increase: i.e., $\mathsf{d}_{\mathrm{ne}}(q, P) \leq \mathsf{d}_{\mathrm{ne}}(q, R)$. This implies that $\delta(q, R) \geq \delta(q, P)$.

Case 2: If $p \notin R$, we can upper-bound q's nearest-neighbor distance in R as follows:

Since R is an α-consistent subset of P, we know that there exists a point $r \in R$ such that $\mathsf{d}(p, r) < \mathsf{d}_{\mathrm{ne}}(p, R)/(1 + \alpha)$. By the triangle inequality and the definition of nearest-enemy, $\mathsf{d}_{\mathrm{ne}}(p, R) \leq \mathsf{d}(p, \mathrm{ne}(q, R)) \leq \mathsf{d}(q, p) + \mathsf{d}_{\mathrm{ne}}(q, R)$. Additionally, applying the definition of chromatic density on q and knowing that $\mathsf{d}_{\mathrm{ne}}(q, P) \leq \mathsf{d}_{\mathrm{ne}}(q, R)$, we have $\mathsf{d}(q, p) = \mathsf{d}_{\mathrm{nn}}(q, P) \leq \mathsf{d}_{\mathrm{nn}}(q, R) = \mathsf{d}_{\mathrm{ne}}(q, R)/(1 + \delta(q, P))$. Therefore:

$$\begin{aligned}\mathsf{d}_{\mathrm{nn}}(q, R) &\leq \mathsf{d}(q, r) \leq \mathsf{d}(q, p) + \mathsf{d}(p, r) \\ &< \mathsf{d}(q, p) + \frac{\mathsf{d}(q, p) + \mathsf{d}_{\mathrm{ne}}(q, R)}{1 + \alpha} \leq \left(\frac{\delta(q, P) + \alpha + 3}{(1 + \alpha)(1 + \delta(q, P))}\right) \mathsf{d}_{\mathrm{ne}}(q, R).\end{aligned}$$

Finally, from the definition of $\delta(q, R)$, we have:

$$\delta(q, R) = \frac{\mathsf{d}_{\mathrm{ne}}(q, R)}{\mathsf{d}_{\mathrm{nn}}(q, R)} - 1 > \frac{(1 + \alpha)(1 + \delta(q, P))}{\delta(q, P) + \alpha + 3} - 1 = \frac{\alpha\,\delta(q, P) - 2}{\delta(q, P) + \alpha + 3}.$$ ◀

The above result can be leveraged to define a coreset, in the sense that an exact result on the coreset corresponds to an approximate result on the original set. As previously defined, we say that a set $R \subseteq P$ is an *ε-coreset for the nearest-neighbor rule* on P, if and only if for every query point $q \in \mathcal{X}$, the class of q's exact nearest-neighbor in R is the same as the class of any of its ε-approximate nearest-neighbors in P.

▶ **Lemma 3.** *Any ε-coreset for the nearest-neighbor rule is an α-consistent subset, for $\alpha \geq 0$.*

Proof. Consider any ε-coreset $C \subseteq P$ for the nearest-neighbor rule on P. Since the approximation guarantee holds for any point in $\mathcal{X}$, it holds for any $p \in P \setminus C$. We know p's nearest-neighbor in the original set P is p itself, thus making $\mathsf{d}_{\mathrm{nn}}(p, P)$ zero. This implies that p must be correctly classified by a nearest-neighbor query on C, that is, $\mathsf{d}_{\mathrm{nn}}(p, C) < \mathsf{d}_{\mathrm{ne}}(p, C)$, which is the definition of α-consistency for any $\alpha \geq 0$. ◀

▶ **Theorem 4.** *Any $2/\varepsilon$-selective subset is an ε-coreset for the nearest-neighbor rule.*

Proof. Let R be an α-selective subset of P, where $\alpha = 2/\varepsilon$. Consider any query point $q \in \mathcal{X}$ in the metric space. It suffices to show that its nearest-neighbor in R is of the same class as any ε-approximate nearest-neighbor in P. To this end, consider q's chromatic density with respect to both P and R, denoted as $\delta(q, P)$ and $\delta(q, R)$, respectively. We identify two cases:

Case 1 (Correct-Classification guarantee): If $\delta(q, P) \geq \varepsilon$.

Consider the bound derived in Lemma 2. Since $\alpha \geq 0$, and by our assumption that $\delta(q, P) \geq \varepsilon > 0$, setting $\alpha = 2/\varepsilon$ implies that $\delta(q, R) > 0$. This means that the nearest-neighbor of q in R belongs to the same class as the nearest-neighbor of q in P. Intuitively, this guarantees that q is correctly classified by the nearest-neighbor rule in R.

Case 2 (ε-Approximation guarantee): If $\delta(q,P) < \varepsilon$.

Let $p \in P$ be q's nearest-neighbor in P, thus $\mathsf{d}(q,p) = \mathsf{d}_{\mathrm{nn}}(q,P)$. Since R is α-selective, there exists a point $r \in R$ such that $\mathsf{d}(p,r) = \mathsf{d}_{\mathrm{nn}}(p,R) < \mathsf{d}_{\mathrm{ne}}(p,P)/(1+\alpha)$. Additionally, by the triangle inequality and the definition of nearest-enemies, we have

$$\mathsf{d}_{\mathrm{ne}}(p,P) \leq \mathsf{d}(p,\mathrm{ne}(q,P)) \leq \mathsf{d}(p,q) + \mathsf{d}(q,\mathrm{ne}(q,P)) = \mathsf{d}_{\mathrm{nn}}(q,P) + \mathsf{d}_{\mathrm{ne}}(q,P).$$

From the definition of chromatic density, $\mathsf{d}_{\mathrm{ne}}(q,P) = (1+\delta(q,P))\,\mathsf{d}_{\mathrm{nn}}(q,P)$. Together, these inequalities imply that $(1+\alpha)\,\mathsf{d}(p,r) \leq (2+\delta(q,P))\,\mathsf{d}_{\mathrm{nn}}(q,P)$. Therefore:

$$\mathsf{d}_{\mathrm{nn}}(q,R) \leq \mathsf{d}(q,r) \leq \mathsf{d}(q,p) + \mathsf{d}(p,r) \leq \left(1 + \frac{2+\delta(q,P)}{1+\alpha}\right)\mathsf{d}_{\mathrm{nn}}(q,P).$$

Now, assuming $\delta(q,P) < \varepsilon$ and setting $\alpha = 2/\varepsilon$, imply that $\mathsf{d}_{\mathrm{nn}}(q,R) < (1+\varepsilon)\,\mathsf{d}_{\mathrm{nn}}(q,P)$. Therefore, the nearest-neighbor of q in R is an ε-approximate nearest-neighbor of q in P.

Cases 1 and 2 imply that setting $\alpha = 2/\varepsilon$ is sufficient to ensure that the nearest-neighbor rule classifies any query point $q \in \mathcal{X}$ with the class of one of its ε-approximate nearest-neighbors in P. Therefore, R is an ε-coreset for the nearest-neighbor rule on P. ◀

So far, we have assumed that nearest-neighbor queries over R are computed exactly, as this is the standard notion of coresets. However, it is reasonable to compute nearest-neighbors approximately even for R. How should the two approximations be combined to achieve a desired final degree of accuracy? Consider another approximation parameter ξ, where $0 \leq \xi < \varepsilon$. We say that a set $R \subseteq P$ is an (ξ,ε)-*coreset* for the approximate nearest-neighbor rule on P, if and only if for every query point $q \in \mathcal{X}$, the class of any of q's ξ-approximate nearest-neighbor in R is the same as the class of any of its ε-approximate nearest-neighbors in P. The following result generalizes Theorem 4 to accommodate for ξ-ANN queries after condensation.

▶ **Theorem 5.** *Any α-selective subset is an (ξ,ε)-coreset for the approximate nearest-neighbor rule when $\alpha = \Omega(1/(\varepsilon-\xi))$.*

Proof. This follows from similar arguments to the ones described in the proof of Theorem 4. Instead, here we set $\alpha = (\varepsilon\xi + 3\xi + 2)/(\varepsilon - \xi)$. Let R be an α-selective subset of P, and $q \in \mathcal{X}$ any query point in the metric space, consider the same two cases:

Case 1 (Correct-Classification guarantee): If $\delta(q,P) \geq \varepsilon$.

Consider the bound derived in Lemma 2. By our assumption that $\delta(q,P) \geq \varepsilon > 0$, and since $\alpha \geq 0$, the following inequality holds true:

$$\delta(q,R) > \frac{\alpha\,\delta(q,P) - 2}{\delta(q,P) + \alpha + 3} \geq \frac{\alpha\varepsilon - 2}{\varepsilon + \alpha + 3}$$

Based on this, it is easy to see that the assignment of $\alpha = (\varepsilon\xi + 3\xi + 2)/(\varepsilon - \xi)$ implies that $\delta(q,R) > \xi$, meaning that any of q's ξ-approximate nearest-neighbors in R belong to the same class as q's nearest-neighbor in P. Intuitively, this guarantees that q is correctly classified by the ξ-ANN rule in R.

Case 2 (ε-Approximation guarantee): If $\delta(q,P) < \varepsilon$.

The assignment of α implies that $\mathsf{d}_{\mathrm{nn}}(q,R) < \frac{1+\varepsilon}{1+\xi}\,\mathsf{d}_{\mathrm{nn}}(q,P)$. This means that an ξ-ANN query for q in R, will return one of q's ε-approximate nearest-neighbors in P.

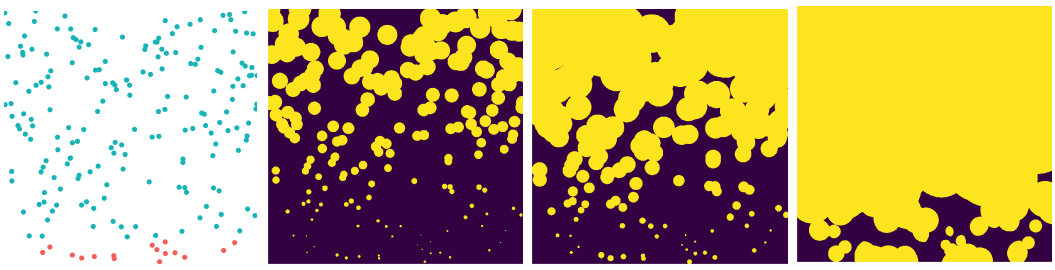
All together, this implies that R is an (ξ,ε)-coreset for the nearest-neighbor rule on P. ◀

In contrast with standard condensation criteria, these new results provide guarantees on either approximation or the correct classification, of any query point in the metric space. This is true even for query points that were "hard" to classify with the entire training set, formally defined as query points with low chromatic density. Consequently, Theorems 4 and 5 show that α must be set to large values if we hope to provide any sort of guarantees for these query points. However, better results can be achieved by restricting the set of points that are guaranteed to be correctly classified. This relates to the notion of *weak* coresets, which provide approximation guarantees only for a subset of the possible queries. Given $\beta \geq 0$, we define $\mathcal{Q}_\beta$ as the set of query points in $\mathcal{X}$ whose chromatic density with respect to P is at least β (i.e., $\mathcal{Q}_\beta = \{q \in \mathcal{X} \mid \delta(q, P) \geq \beta\}$). The following result describes the trade-off between α and β to guarantee the correct classification of query points in $\mathcal{Q}_\beta$ after condensation.

▶ **Theorem 6.** *Any α-consistent subset is a weak ε-coreset for the nearest-neighbor rule for queries in $\mathcal{Q}_\beta$, for $\beta = 2/\alpha$. Moreover, all query points in $\mathcal{Q}_\beta$ are correctly classified.*

The proof of this theorem is rather simple, and follows the same arguments outlined in Case 1 of the proof of Theorem 4. Basically, we use Lemma 2 to show that for any query point $q \in \mathcal{Q}_\beta$, q's chromatic density after condensation is greater than zero if $\alpha\beta \geq 2$. Note that ε plays no role in this result, as the guarantee on query points of $\mathcal{Q}_\beta$ is of correct classification (i.e., the class of its *exact* nearest-neighbor in P), rather than an approximation.

The trade-off between α and β is illustrated in Figure 3. From an initial training set $P \subset \mathbb{R}^2$ (Figure 3a), we show the regions of $\mathbb{R}^2$ that comprise the sets $\mathcal{Q}_\beta$ for $\beta = 2/\alpha$, using $\alpha = \{0.1, 0.2, \sqrt{2}\}$ (Figures 3b-3d). While evidently, increasing α guarantees that more query points will be correctly classified after condensation, this example demonstrates a phenomenon commonly observed experimentally: most query points lie far from enemy points, and thus have high chromatic density with respect to P. Therefore, while Theorem 4 states that α must be set to $2/\varepsilon$ to provide approximation guarantees on all query points, Theorem 6 shows that much smaller values of α are sufficient to provide guarantees on some query points, as evidenced in the example in Figure 3.



(a) Training set (200 pts). **(b)** $\mathcal{Q}_{2/\alpha}$ for $\alpha = 0.1$. **(c)** $\mathcal{Q}_{2/\alpha}$ for $\alpha = 0.2$. **(d)** $\mathcal{Q}_{2/\alpha}$ for $\alpha = \sqrt{2}$.

Figure 3 Depiction of the $\mathcal{Q}_\beta$ sets for which any α-consistent subset is weak coreset ($\beta = 2/\alpha$). Query points in the *yellow* • areas are inside $\mathcal{Q}_\beta$, and thus correctly classified after condensation. Query points in the *blue* • areas are not in $\mathcal{Q}_\beta$, and have no guarantee of correct classification.

These results establish a clear connection between the problem of condensation and that of finding coresets for the nearest-neighbor rule, and provides a roadmap to prove Theorem 1. This is the first characterization of sufficient conditions to correctly classify any query point in $\mathcal{X}$ after condensation, and not just the points in P (as the original consistency criteria implies). In the following section, these existential results are matched with algorithms to compute α-selective subsets of P of bounded cardinality.

3 Coreset Computation

3.1 Hardness Results

Define Min-α-CS to be the problem of computing an α-consistent subset of minimum cardinality for a given training set P. Similarly, let Min-α-SS be the corresponding optimization problem for α-selective subsets. Following known results from standard condensation [28, 39, 40], when α is set to zero, the Min-0-CS and Min-0-SS problems are both known to be NP-hard. Being special cases of the general problems just defined, this implies that both Min-α-CS and Min-α-SS are NP-hard.

In this section, we present results related to the hardness of approximation of both problems, along with simple algorithmic approaches with tight approximation factors.

▶ **Theorem 7.** *The Min-α-CS problem is* NP*-hard to approximate in polynomial time within a factor of* $2^{(\mathrm{ddim}(\mathcal{X})\log((1+\alpha)/\gamma))^{1-o(1)}}$.

The full proof is omitted, as it follows from a modification of the hardness bounds proof for the Min-0-CS problem described in [22], which is based on a reduction from the *Label Cover* problem. Proving Theorem 7 involves a careful adjustment of the distances in this reduction, so that all the points in the construction have chromatic density at least α. Consequently, this implies that the minimum nearest-enemy distance is reduced by a factor of $1/(1+\alpha)$, explaining the resulting bound for Min-α-CS.

The NET algorithm [22] can also be generalized to compute α-consistent subsets of P as follows. We define α-NET as the algorithm that computes a $\gamma/(1+\alpha)$-net of P, where γ is the smallest nearest-enemy distance in P. The covering property of nets [24] implies that the resulting subset is α-consistent, while the packing property suggests that its cardinality is $\mathcal{O}\left(((1+\alpha)/\gamma)^{\mathrm{ddim}(\mathcal{X})+1}\right)$, implying a tight approximation to the Min-α-CS problem.

▶ **Theorem 8.** *The Min-α-SS problem is* NP*-hard to approximate in polynomial time within a factor of* $(1-o(1))\ln n$ *unless* NP $\subseteq$ DTIME$(n^{\log\log n})$.

Proof. The result follows from the hardness of another related covering problem: the minimum *dominating set* [16, 30, 32]. We describe a simple L-reduction from any instance of this problem to an instance of Min-α-SS, which preserves the approximation ratio.

1. Consider any instance of minimum dominating set, consisting of the graph $G = (V, E)$.
2. Generate a new edge-weighted graph G' as follows:
   Create two copies of G, namely $G_{\mathsf{r}} = (V_{\mathsf{r}}, E_{\mathsf{r}})$ and $G_{\mathsf{b}} = (V_{\mathsf{b}}, E_{\mathsf{b}})$, of *red* and *blue* nodes respectively. Set all edge-weights of G_{r} and G_{b} to be 1. Finally, connect each red node v_{r} to its corresponding blue node v_{b} by an edge $\{v_{\mathsf{r}}, v_{\mathsf{b}}\}$ of weight $1+\alpha+\xi$ for a sufficiently small constant $\xi > 0$. Formally, G' is defined as the edge-weighted graph $G' = (V', E')$ where the set of nodes is $V' = V_{\mathsf{r}} \cup V_{\mathsf{b}}$, the set of edges is $E' = E_{\mathsf{r}} \cup E_{\mathsf{r}} \cup \{\{v_{\mathsf{r}}, v_{\mathsf{b}}\} \mid v \in V\}$, and an edge-weight function $w : E' \to \mathbb{R}^+$ where $w(e) = 1$ iff $e \in E_{\mathsf{r}} \cup E_{\mathsf{b}}$, and $w(e) = 1+\alpha+\xi$ otherwise.
3. A labeling function l where $l(v) = \textit{red}$ iff $v \in V_{\mathsf{r}}$, and $l(v) = \textit{blue}$ iff $v \in V_{\mathsf{b}}$.
4. Compute the shortest-path metric of G', denoted as $\mathsf{d}_{G'}$.
5. Solve the Min-α-SS problem for the set V', on metric $\mathsf{d}_{G'}$, and the labels defined by l.

A dominating set of G consists of a subset of nodes $D \subseteq V$, such that every node $v \in V \setminus D$ is adjacent to a node in D. Given any dominating set $D \subseteq V$ of G, it is easy to see that the subset $R = \{v_{\mathsf{r}}, v_{\mathsf{b}} \mid v \in D\}$ is an α-selective subset of V', where $|R| = 2|D|$.

Similarly, given an α-selective subset $R \subseteq V'$, there is a corresponding dominating set D of G, where $|D| \leq |R|/2$, as D can be either $R \cap V_{\mathsf{r}}$ or $R \cap V_{\mathsf{b}}$. Therefore, MIN-α-SS is as hard to approximate as the minimum dominating set problem. ◀

There is a clear connection between the MIN-α-SS problem and covering problems, in particular that of finding an optimal hitting set. Given a set of elements U and a family C of subsets of U, a *hitting set* of (U, C) is a subset $H \subseteq U$ such that every set in C contains at least one element of H. Therefore, let $N_{p,\alpha}$ be the set of points of P whose distance to p is less than $\mathsf{d}_{\mathrm{ne}}(p)/(1+\alpha)$, then any hitting set of $(P, \{N_{p,\alpha} \mid p \in P\})$ is also an α-selective subset of P, and vice versa. This simple reduction implies a $\mathcal{O}(n^3)$ worst-case time $\mathcal{O}(\log n)$-approximation algorithm for MIN-α-SS, based on the classic greedy algorithm for set cover [12, 35]. Call this approach α-HSS or *α-Hitting Selective Subset*. It follows from Theorem 8 that for training sets in general metric spaces, this is the best approximation possible under standard complexity assumptions.

While both α-NET and α-HSS compute tight approximations of their corresponding problems, their performance in practice does not compare to heuristic approaches for standard condensation (see Section 4 for experimental results). Therefore, in the next section, we consider one such heuristic and extend it to compute subsets with the newly defined criteria.

3.2 A Practical Algorithm

For standard condensation, the RSS algorithm was recently proposed [20] to compute selective subsets. It runs in quadratic worst-case time and exhibits good performance in practice. The selection process of this algorithm is heuristic in nature and can be described as follows: beginning with an empty set, the points in $p \in P$ are examined in increasing order with respect to their nearest-enemy distance $\mathsf{d}_{\mathrm{ne}}(p)$. The point p is added to the subset R if $\mathsf{d}_{\mathrm{nn}}(p, R) \geq \mathsf{d}_{\mathrm{ne}}(p)$. It is easy to see that the resulting subset is selective.

We define a generalization, called α-RSS, to compute α-selective subsets of P. The condition to add a point $p \in P$ to the selected subset checks if any previously selected point is closer to p than $\mathsf{d}_{\mathrm{ne}}(p)/(1+\alpha)$, instead of just $\mathsf{d}_{\mathrm{ne}}(p)$. See Algorithm 1 for a formal description, and Figure 4 for an illustration. It is easy to see that this algorithm computes an α-selective subset, while keeping the quadratic time complexity of the original RSS algorithm.

Algorithm 1 α-RSS.

Input: Initial training set P and parameter $\alpha \geq 0$
Output: α-selective subset $R \subseteq P$

```
R ← φ
Let {p_i}_{i=1}^n be the points of P sorted increasingly w.r.t. d_ne(p_i)
foreach p_i ∈ P, where i = 1...n do
    if d_nn(p_i, R) ≥ d_ne(p_i)/(1+α) then
        R ← R ∪ {p_i}
return R
```

Naturally, we want to analyze the number of points this algorithm selects. The remainder of this section establishes upper-bounds and approximation guarantees of the α-RSS algorithm for any doubling metric space, with improved results in the Euclidean space. This resolves the open problem posed in [20] of whether RSS computes an approximation of the MIN-0-CS and MIN-0-SS problems.

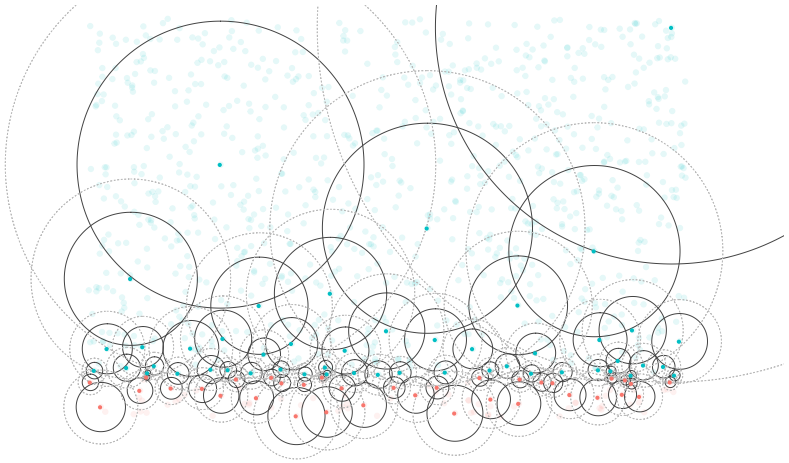


Figure 4 Selection of α-RSS for α=0.5. Faded points are not selected, while selected points are drawn along with a ball of radius $\mathsf{d}_{\mathrm{ne}}(p)$ (dotted outline) and a ball of radius $\mathsf{d}_{\mathrm{ne}}(p)/(1+\alpha)$ (solid outline). A point p is selected if no previously selected point is closer to p than $\mathsf{d}_{\mathrm{ne}}(p)/(1+\alpha)$.

Size in Doubling spaces. First, we consider the case where the underlying metric space $(\mathcal{X}, \mathsf{d})$ of P is doubling. The following results depend on the doubling dimension $\mathrm{ddim}(\mathcal{X})$ of the metric space (which is assumed to be constant), the margin γ (the smallest nearest-enemy distance of any point in P), and κ (the number of nearest-enemy points in P).

▶ **Theorem 9.** *α-RSS computes a tight approximation for the Min-α-CS problem.*

Proof. This follows from a direct comparison to the resulting subset of the α-NET algorithm from the previous section. For any point p selected by α-NET, let $B_{p,\alpha}$ be the set of points of P "covered" by p, that is, whose distance to p is at most $\gamma/(1+\alpha)$. By the covering property of ε-nets, this defines a partition on P when considering every point p selected by α-NET.

Let R be the set of points selected by α-RSS, we analyze the size of $B_{p,\alpha} \cap R$, that is, for any given $B_{p,\alpha}$ how many points could have been selected by the α-RSS algorithm. Let $a, b \in B_{p,\alpha} \cap R$ be any two such points, where without loss of generality, $\mathsf{d}_{\mathrm{ne}}(a) \leq \mathsf{d}_{\mathrm{ne}}(b)$. By the selection process of the algorithm, we know that $\mathsf{d}(a,b) \geq \mathsf{d}_{\mathrm{ne}}(b)/(1+\alpha) \geq \gamma/(1+\alpha)$. A simple packing argument in doubling metrics implies that $|B_{p,\alpha} \cap R| \leq 2^{\mathrm{ddim}(\mathcal{X})+1}$. Altogether, we have that the size of the subset selected by α-RSS is $\mathcal{O}\left((2(1+\alpha)/\gamma)^{\mathrm{ddim}(\mathcal{X})+1}\right)$. ◀

▶ **Theorem 10.** *α-RSS computes an $\mathcal{O}\left(\log\left(\min\left(1+2/\alpha, 1/\gamma\right)\right)\right)$-factor approximation for the Min-α-SS problem. For $\alpha = \Omega(1)$, this is a constant-factor approximation.*

Proof. Let OPT_α be the optimum solution to the Min-α-SS problem, i.e., the minimum cardinality α-selective subset of P. For every point $p \in \mathrm{OPT}_\alpha$ in such solution, define $S_{p,\alpha}$ to be the set of points in P "covered" by p, or simply $S_{p,\alpha} = \{r \in P \mid \mathsf{d}(r,p) < \mathsf{d}_{\mathrm{ne}}(r)/(1+\alpha)\}$. Additionally, let R be the set of points selected by α-RSS, define $R_{p,\sigma}$ to be the points selected by α-RSS which also belong to $S_{p,\alpha}$ and whose nearest-enemy distance is between σ and 2σ, for $\sigma \in [\gamma, 1]$. That is, $R_{p,\sigma} = \{r \in R \cap S_{p,\alpha} \mid \mathsf{d}_{\mathrm{ne}}(r) \in [\sigma, 2\sigma)\}$. Clearly, these subsets define a partitioning of R for all $p \in \mathrm{OPT}_\alpha$ and values of $\sigma = \gamma\, 2^i$ for $i = \{0, 1, 2, \ldots, \lceil \log \frac{1}{\gamma} \rceil\}$.

However, depending on α, some values of σ would yield empty $R_{p,\sigma}$ sets. Consider some point $q \in S_{p,\alpha}$, we can bound its nearest-enemy distance with respect to the nearest-enemy distance of point p. In particular, by leveraging simple triangle-inequality arguments, it is possible to prove that $\frac{1+\alpha}{2+\alpha}\,\mathsf{d}_{\mathrm{ne}}(p) \leq \mathsf{d}_{\mathrm{ne}}(q) \leq \frac{1+\alpha}{\alpha}\,\mathsf{d}_{\mathrm{ne}}(p)$. Therefore, the values of σ for which $R_{p,\sigma}$ sets are not empty, are $\sigma = 2^j\,\frac{1+\alpha}{2+\alpha}\,\mathsf{d}_{\mathrm{ne}}(p)$ for $j = \{0, \ldots, \lceil \log(1 + 2/\alpha) \rceil\}$.

The proof now follows by bounding the size of $R_{p,\sigma}$ which can be achieved by bounding its spread. Thus, lets consider the smallest and largest pairwise distances among points in $R_{p,\sigma}$. Take any two points $a, b \in R_{p,\sigma}$ where without loss of generality, $\mathsf{d}_{\mathrm{ne}}(a) \leq \mathsf{d}_{\mathrm{ne}}(b)$. Note that points selected by α-RSS cannot be "too close" to each other; that is, as a and b were selected by the algorithm, we know that $(1+\alpha)\,\mathsf{d}(a,b) \geq \mathsf{d}_{\mathrm{ne}}(b) \geq \sigma$. Therefore, the smallest pairwise distance in $R_{p,\sigma}$ is at least $\sigma/(1+\alpha)$. Additionally, by the triangle inequality, we can bound the maximum pairwise distance using their distance to p as $\mathsf{d}(a,b) \leq \mathsf{d}(a,p) + \mathsf{d}(p,b) \leq 4\sigma/(1+\alpha)$. Then, by the packing properties of doubling spaces, the size of $R_{p,\sigma}$ is at most $4^{\mathrm{ddim}(\mathcal{X})+1}$.

Altogether, for every $p \in \mathrm{OPT}_\alpha$ there are up to $\lceil \log(\min(1 + 2/\alpha, 1/\gamma)) \rceil$ non-empty $R_{p,\sigma}$ subsets, each containing at most $4^{\mathrm{ddim}(\mathcal{X})+1}$ points. In doubling spaces with constant doubling dimension, the size of these subsets is also constant. ◀

While these results are meaningful from a theoretical perspective, it is also useful to establishing bounds in terms of the geometry of the learning space, which is characterized by the boundaries between points of different classes. Thus, using similar packing arguments as above, we bound the selection size of the algorithm with respect to κ.

▶ **Theorem 11.** α-RSS *selects* $\mathcal{O}\left(\kappa \log \frac{1}{\gamma}\ (1+\alpha)^{\mathrm{ddim}(\mathcal{X})+1}\right)$ *points.*

Proof. This follows from similar arguments to the ones used to prove Theorem 10, using an alternative charging scheme for each nearest-enemy point in the training set. Consider one such point $p \in \{\mathrm{ne}(r) \mid r \in P\}$ and a value $\sigma \in [\gamma, 1]$, we define $R'_{p,\sigma}$ to be the subset of points from α-RSS whose nearest-enemy is p, and their nearest-enemy distance is between σ and 2σ. That is, $R'_{p,\sigma} = \{r \in R \mid \mathrm{ne}(r) = p \wedge \mathsf{d}_{\mathrm{ne}}(r) \in [\sigma, 2\sigma)\}$. These subsets partition R for all nearest-enemy points of P, and values of $\sigma = \gamma\, 2^i$ for $i = \{0, 1, 2, \ldots, \lceil \log \frac{1}{\gamma} \rceil\}$.

For any two points $a, b \in R'_{p,\sigma}$, the selection criteria of α-RSS implies some separation between selected points, which can be used to prove that $\mathsf{d}(a,b) \geq \sigma/(1+\alpha)$. Additionally, we know that $\mathsf{d}(a,b) \leq \mathsf{d}(a,p) + \mathsf{d}(p,b) = \mathsf{d}_{\mathrm{ne}}(a) + \mathsf{d}_{\mathrm{ne}}(b) \leq 4\sigma$. Using a simple packing argument, we have that $|R'_{p,\sigma}| \leq \lceil 4(1+\alpha) \rceil^{\mathrm{ddim}(\mathcal{X})+1}$.

Altogether, by counting all sets $R'_{p,\sigma}$ for each nearest-enemy in the training set and values of σ, the size of R is upper-bounded by $|R| \leq \kappa \lceil \log 1/\gamma \rceil \lceil 4(1+\alpha) \rceil^{\mathrm{ddim}(\mathcal{X})+1}$. Based on the assumption that $\mathrm{ddim}(\mathcal{X})$ is constant, this completes the proof. ◀

As a corollary, this result implies that when $\alpha = 2/\varepsilon$, the α-selective subset computed by α-RSS contains $\mathcal{O}\left(\kappa \log 1/\gamma\ (1/\varepsilon)^{\mathrm{ddim}(\mathcal{X})+1}\right)$ points. This establishes the size bound on the ε-coreset given in Theorem 1, which can be computed using the α-RSS algorithm.

Size in Euclidean space. In the case where $P \subset \mathbb{R}^d$ lies in d-dimensional Euclidean space, the analysis of α-RSS can be further improved, leading to a constant-factor approximation of Min-α-SS for any value of $\alpha \geq 0$, and reduced dependency on the dimensionality of P.

▶ **Theorem 12.** α-RSS *computes an* $\mathcal{O}(1)$*-approximation for the* Min-α*-SS problem in* $\mathbb{R}^d$.

Proof. Similar to the proof of Theorem 10, define $R_p = S_{p,\alpha} \cap R$ as the points selected by α-RSS that are "covered" by p in the optimum solution OPT_α. Consider two such points $a, b \in R_p$ where without loss of generality, $\mathsf{d}_{\mathrm{ne}}(a) \leq \mathsf{d}_{\mathrm{ne}}(b)$. By the definition of $S_{p,\alpha}$ we know that $\mathsf{d}(a,p) < \mathsf{d}_{\mathrm{ne}}(a)/(1+\alpha)$, and similarly with b. Additionally, from the selection of the algorithm we know that $\mathsf{d}(a,b) \geq \mathsf{d}_{\mathrm{ne}}(b)/(1+\alpha)$. Overall, these inequalities imply that the angle $\angle apb \geq \pi/3$. By a simple packing argument, the size of R_p is bounded by the *kissing number* in d-dimensional Euclidean space, or simply $\mathcal{O}((3/\pi)^{d-1})$. Therefore, we have that $|R| \leq \sum_p |R_p| = |\mathrm{OPT}_\alpha|\ \mathcal{O}((3/\pi)^{d-1})$. Assuming d is constant, this completes the proof. ◀

Furthermore, a similar constant-factor approximation can be achieved for any training set P in ℓ_p space for $p \geq 3$. This follows analogously to the proof of Theorem 12, exploiting the bounds between ℓ_p and ℓ_2 metrics, where $1/\sqrt{d}\ \|v\|_p \leq \|v\|_2 \leq \sqrt{d}\ \|v\|_p$. This would imply that the angle between any two points in α-RSS_p is $\Omega(1/d)$. Therefore, it shows that α-RSS achieves an approximation factor of $\mathcal{O}(d^{d-1})$, or simply $\mathcal{O}(1)$ for constant dimension.

Similarly to the case of doubling spaces, we also establish upper-bounds in terms of κ for the selection size of the algorithm in Euclidean space. The following result improves the exponential dependence on the dimensionality of P (from $\mathsf{ddim}(\mathbb{R}^d) = \Theta(d)$ to $d-1$), while keeping the dependency on the margin γ, which contrast with the approximation factor results.

▶ **Theorem 13.** *In Euclidean space $\mathbb{R}^d$, α-RSS selects $\mathcal{O}\left(\kappa \log \frac{1}{\gamma}\ (1+\alpha)^{d-1}\right)$ points.*

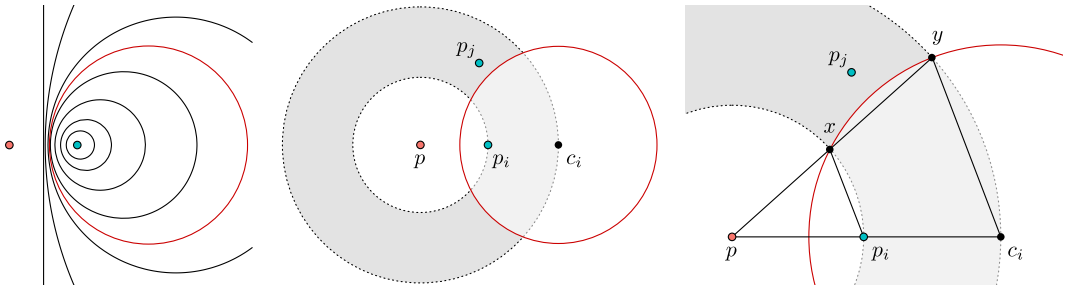
Proof. Let p be any nearest-enemy point of P and $\sigma \in [\gamma, 1]$, similarly define $R'_{p,\sigma}$ to be the set of points selected by α-RSS whose nearest-enemy is p and their nearest-enemy distance is between σ and $b\sigma$, for $b = \frac{(1+\alpha)^2}{\alpha(2+\alpha)}$. Equivalently, these subsets define a partitioning of R for all nearest-enemy points p and values of $\sigma = \gamma\, b^k$ for $k = \{0, 1, 2, \ldots, \lceil \log_b \frac{1}{\gamma} \rceil\}$. Thus, the proof follows from bounding the minimum angle between points in these subsets. For any two such points $p_i, p_j \in R'_{p,\sigma}$, we lower bound the angle $\angle p_i p p_j$. Assume without loss of generality that $\mathsf{d}_{\mathrm{ne}}(p_i) \leq \mathsf{d}_{\mathrm{ne}}(p_j)$. By definition of the partitioning, we also know that $\mathsf{d}_{\mathrm{ne}}(p_j) \leq b\sigma \leq b\, \mathsf{d}_{\mathrm{ne}}(p_i)$. Therefore, altogether we have that $\mathsf{d}_{\mathrm{ne}}(p_i) \leq \mathsf{d}_{\mathrm{ne}}(p_j) \leq b\, \mathsf{d}_{\mathrm{ne}}(p_i)$.

First, consider the set of points whose distance to p_i is $(1+\alpha)$ times their distance to p, which defines a multiplicative weighted bisector [6] between points p and p_i, with weights equal to 1 and $1/(1+\alpha)$ respectively. This is characterized as a d-dimensional ball (see Figure 5a) with center $c_i = (p_i - p)\, b + p$ and radius $\mathsf{d}_{\mathrm{ne}}(p_i)\, b/(1+\alpha)$. Thus p, p_i and c_i are collinear, and the distance between p and c_i is $\mathsf{d}(p, c_i) = b\, \mathsf{d}_{\mathrm{ne}}(p_i)$. In particular, let's consider the relation between p_j and such bisector. As p_j was selected by the algorithm after p_i, we know that $(1+\alpha)\, \mathsf{d}(p_j, p_i) \geq \mathsf{d}_{\mathrm{ne}}(p_j)$ where $\mathsf{d}_{\mathrm{ne}}(p_j) = \mathsf{d}(p_j, p)$. Therefore, clearly p_j lies either outside or in the surface of the weighted bisector between p and p_i (see Figure 5b).

For angle $\angle p_i p p_j$, we can frame the analysis to the plane defined by p, p_i and p_j. Let x and y be two points in this plane, such that they are the intersection points between the weighted bisector and the balls centered at p of radii $\mathsf{d}_{\mathrm{ne}}(p_i)$ and $b\, \mathsf{d}_{\mathrm{ne}}(p_i)$ respectively (see Figure 5c). By the convexity of the weighted bisector between p and p_i, we can say that $\angle p_i p p_j \geq \min(\angle xpp_i, \angle ypc_j)$. Now, consider the triangles $\triangle pxp_i$ and $\triangle pyc_i$. By the careful selection of b, these triangles are both isosceles and similar. In particular, for $\triangle pxp_i$ the two sides incident to p have length equal to $\mathsf{d}_{\mathrm{ne}}(p_i)$, and the side opposite to p has length equal to $\mathsf{d}_{\mathrm{ne}}(p_i)/(1+\alpha)$. For $\triangle pyc_i$, the side lengths are $b\, \mathsf{d}_{\mathrm{ne}}(p_i)$ and $\mathsf{d}_{\mathrm{ne}}(p_i)\, b/(1+\alpha)$. Therefore, the angle $\angle p_i p p_j \geq \angle xpp_i \geq 1/(1+\alpha)$.

By a simple packing argument based on this minimum angle, we have that the size of $R'_{p,\sigma}$ is $\mathcal{O}((1+\alpha)^{d-1})$. All together, following the defined partitioning, we have that:

$$|R| = \sum_p \sum_{k=0}^{\lceil \log_b \frac{1}{\gamma} \rceil} |R'_{p,b^k}| \leq \kappa \left\lceil \log_b \frac{1}{\gamma} \right\rceil \mathcal{O}\left((1+\alpha)^{d-1}\right)$$



(a) Multiplicatively weighted bisectors for different weights.

(b) Position of point p_j w.r.t. the weighted bisector between points p and p_i.

(c) The intersection points x and y between the weighted bisector and the limit balls of $R_{p,\sigma}$.

Figure 5 Construction for the analysis of the minimum angle between two points in $R'_{p,\sigma}$ w.r.t. some nearest-enemy point $p \in P$. Let points $p_i, p_j \in R'_{p,\sigma}$, we analyze the angle $\angle p_i p p_j$.

For constant α and d, the size of α-RSS is $\mathcal{O}(\kappa \log \frac{1}{\gamma})$. Moreover, when α is zero α-RSS selects $\mathcal{O}(\kappa\ c^{d-1})$, matching the previously known bound for RSS in Euclidean space. ◀

3.3 Subquadratic Algorithm

In this section we present a subquadratic implementation for the α-RSS algorithm, which completes the proof of our main result, Theorem 1. Among algorithms for nearest-neighbor condensation, FCNN achieves the best worst-case time complexity, running in $\mathcal{O}(nm)$ time, where $m = |R|$ is the size of the selected subset.

The α-RSS algorithm consists of two main stages: computing the nearest-enemy distances of all points in P (and sorting the points based on these), and the selection process itself. The first stage requires a total of n nearest-enemy queries, plus additional $\mathcal{O}(n \log n)$ time for sorting. The second stage performs n nearest-neighbor queries on the current selected subset R, which needs to be updated m times. In both cases, using exact nearest-neighbor search would degenerate into linear search due to the *curse of dimensionality*. Thus, the first and second stage of the algorithm would need $\mathcal{O}(n^2)$ and $\mathcal{O}(nm)$ worst-case time respectively.

These bottlenecks can be overcome by leveraging approximate nearest-neighbor techniques. Clearly, the first stage of the algorithm can be improved by computing nearest-enemy distances approximately, using as many ANN structures as classes there are in P, which is considered to be a small constant. Therefore, by also applying a simple brute-force search for nearest-neighbors in the second stage, result (i) of the next theorem follows immediately. Moreover, by combining this with standard techniques for static-to-dynamic conversions [9], we have result (ii) below. Denote this variant of α-RSS as (α, ξ)-RSS, for a parameter $\xi \geq 0$.

▶ **Theorem 14.** *Given a data structure for ξ-ANN searching with construction time t_c and query time t_q (which potentially depend on n and ξ), the (α, ξ)-RSS variant can be implemented with the following worst-case time complexities, where m is the size of the selected subset.*

(i) $\mathcal{O}\left(t_c + n\left(t_q + m + \log n\right)\right)$

(ii) $\mathcal{O}\left(\left(t_c + n\, t_q\right) \log n\right)$

More generally, if we are given an additional data structure for dynamic ξ-ANN searching with construction time t'_c, query time t'_q, and insertion time t'_i, the overall running time will be $\mathcal{O}\left(t_c + t'_c + n\left(t_q + t'_q + \log n\right) + m\, t'_i\right)$. Indeed, this can be used to obtain (ii) from the static-to-dynamic conversions [9], which propose an approach to convert static search

structures into dynamic ones. These results directly imply implementations of (α, ξ)-RSS with subquadratic worst-case time complexities, based on ANN techniques [4, 5] for low-dimensional Euclidean space, and using techniques like LSH [2] that are suitable for ANN in high-dimensional Hamming and Euclidean spaces. More generally, subquadratic runtimes can be achieved by leveraging techniques [13] for dynamic ANN search in doubling spaces.

Dealing with uncertainty. Such implementation schemes for α-RSS would incur an approximation error (of up to $1+\xi$) on the computed distances: either only during the first stage if (i) is implemented, or during both stages if (ii) or the dynamic-structure scheme are implemented. The uncertainty introduced by these approximate queries, imply that in order to guarantee finding α-selective subsets, we must modify the condition for adding point during the second stage of the algorithm. Let $\mathsf{d}_{\mathrm{ne}}(p, \xi)$ denote the ξ-approximate nearest-enemy distance of p computed in the first stage, and let $\mathsf{d}_{\mathrm{nn}}(p, R, \xi)$ denote the ξ-approximate nearest-neighbor distance of p over points of the current subset (computed in the second stage). Then, (α, ξ)-RSS adds a point p into the subset if $(1+\xi)(1+\alpha)\,\mathsf{d}_{\mathrm{nn}}(p, R, \xi) \geq \mathsf{d}_{\mathrm{ne}}(p, \xi)$.

By similar arguments to the ones described in Section 3.2, size guarantees can be extended to (α, ξ)-RSS. First, the size of the subset selected by (α, ξ)-RSS, in terms of the number of nearest-enemy points in the set, would be bounded by the size of the subset selected by $\hat{\alpha}$-RSS with $\hat{\alpha} = (1+\alpha)(1+\xi)^2 - 1$. Additionally, the approximation factor of (α, ξ)-RSS in both doubling and Euclidean metric spaces would increase by a factor of $\mathcal{O}((1+\xi)^{2(\mathrm{ddim}(\mathcal{X})+1)})$.

This completes the proof of Theorem 1.

4 Experimental Evaluation

In order to get a clearer impression of the relevance of these results in practice, we performed experimental trials on several training sets, both synthetically generated and widely used benchmarks. First, we consider 21 training sets from the UCI *Machine Learning Repository*[2] which are commonly used in the literature to evaluate condensation algorithms [21]. These consist of a number of points ranging from 150 to 58000, in d-dimensional Euclidean space with d between 2 and 64, and 2 to 26 classes. We also generated some synthetic training sets, containing 10^5 uniformly distributed points, in 2 to 3 dimensions, and 3 classes. All training sets used in these experimental trials are summarized in Table 1. The implementation of the algorithms, training sets used, and raw results, are publicly available[3].

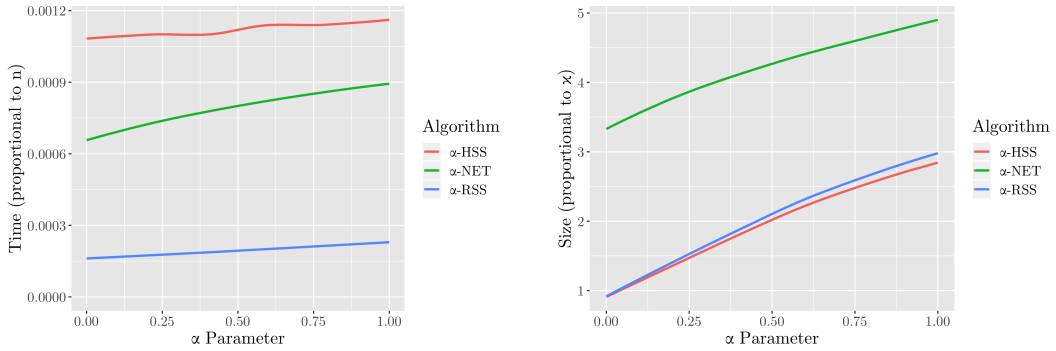
These experimental trials compare the performance of different condensation algorithms when applied to vastly different training sets. We use two measures of comparison on these algorithms: their runtime in the different training sets, and the size of the subset selected. Clearly, these values might differ greatly on training sets whose size are too distinct. Therefore, before comparing the raw results, these are normalized. The runtime of an algorithm for a given training set is normalized by dividing it by n, the size of the training set. The size of the selected subset is normalized by dividing it by κ, the number of nearest-enemy points in the training set, which characterizes the complexity of the boundaries between classes.

Algorithm Comparison. The first experiment evaluates the performance of the three algorithms proposed in this paper: α-RSS, α-HSS, and α-NET. The evaluation is carried out by varying the value of the α parameter from 0 to 1, to understand the impact of increasing

[2] `https://archive.ics.uci.edu/ml/index.php`
[3] `https://github.com/afloresv/nnc/`

this parameter. The implementation of α-HSS uses the well-known greedy algorithm for set cover [12], and solves the problem using the reduction described in Section 3.1. In the other hand, recall that the original NET algorithm (for $\alpha = 0$) implements an extra pruning technique to further reduce the training set after computing the γ-net. To do a fair comparison between the techniques, we implemented the α-NET algorithm with a modified version of this pruning techinque that guarantees that the selected subset is still α-selective.

The results show that α-RSS outperforms the other algorithms in terms of running time by a big margin, and irrespective of the value of α (see Figure 6a). Additionally, the number of points selected by α-RSS is comparable to α-HSS, which guarantees the best possible approximation factor in general metrics, while α-NET is significantly outperformed.

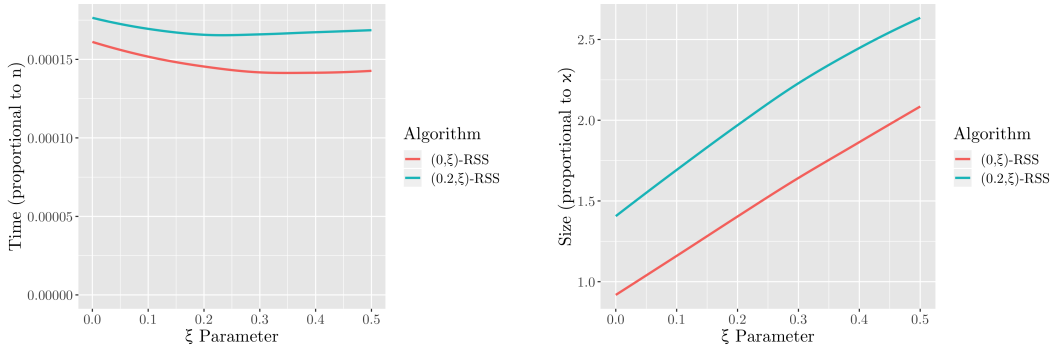


(a) Running time.

(b) Size of the selected subsets.

Figure 6 Comparison α-RSS, α-NET, and α-HSS, for different values of α.

Subquadratic Approach. Using the same experimental framework, we evaluate performance of the subquadratic implementation (α, ξ)-RSS described in Section 3.3. In this case, we change the value of parameter ξ to assess its effect on the running time and selection size over the algorithm, for two different values of α (see Figure 7). The results show an expected increase of the number of selected points, while significantly improving its running time.



(a) Running time.

(b) Size of the selected subsets.

Figure 7 Evaluating the effect of increasing the parameter ξ on (α, ξ)-RSS for $\alpha = \{0, 0.2\}$.

Table 1 Training sets used to evaluate the performance of condensation algorithms. Indicates the number of points n, dimensions d, classes c, nearest-enemy points κ (also in percentage *w.r.t.* n).

Training set	n	d	c	κ (%)
banana	5300	2	2	811 (15.30%)
cleveland	297	13	5	125 (42.09%)
glass	214	9	6	87 (40.65%)
iris	150	4	3	20 (13.33%)
iris2d	150	2	3	13 (8.67%)
letter	20000	16	26	6100 (30.50%)
magic	19020	10	2	5191 (27.29%)
monk	432	6	2	300 (69.44%)
optdigits	5620	64	10	1245 (22.15%)
pageblocks	5472	10	5	429 (7.84%)
penbased	10992	16	10	1352 (12.30%)
pima	768	8	2	293 (38.15%)
ring	7400	20	2	2369 (32.01%)
satimage	6435	36	6	1167 (18.14%)
segmentation	2100	19	7	398 (18.95%)
shuttle	58000	9	7	920 (1.59%)
thyroid	7200	21	3	779 (10.82%)
twonorm	7400	20	2	1298 (17.54%)
wdbc	569	30	2	123 (21.62%)
wine	178	13	3	37 (20.79%)
wisconsin	683	9	2	35 (5.12%)
v-100000-2-3-15	100000	2	3	1909 (1.90%)
v-100000-2-3-5	100000	2	3	788 (0.78%)
v-100000-3-3-15	100000	3	3	7043 (7.04%)
v-100000-3-3-5	100000	3	3	3738 (3.73%)
v-100000-4-3-15	100000	4	3	13027 (13.02%)
v-100000-4-3-5	100000	4	3	10826 (10.82%)
v-100000-5-3-15	100000	5	3	22255 (22.25%)
v-100000-5-3-5	100000	5	3	17705 (17.70%)

References

1 Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

2 Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. Approximate nearest neighbor search in high dimensions. *arXiv preprint*, 2018. `arXiv:1806.09823`.

3 Fabrizio Angiulli. Fast nearest neighbor condensation for large data sets classification. *IEEE Transactions on Knowledge and Data Engineering*, 19(11):1450–1464, 2007.

4 Sunil Arya, Guilherme D Da Fonseca, and David M Mount. Approximate polytope membership queries. *SIAM Journal on Computing*, 47(1):1–51, 2018.

5 Sunil Arya, Theocharis Malamatos, and David M Mount. Space-time tradeoffs for approximate nearest neighbor searching. *Journal of the ACM (JACM)*, 57(1):1, 2009.

6 Franz Aurenhammer and Herbert Edelsbrunner. An optimal algorithm for constructing the weighted voronoi diagram in the plane. *Pattern Recognition*, 17(2):251–257, 1984.

7 Ricardo Barandela, Francesc J Ferri, and J Salvador Sánchez. Decision boundary preserving prototype selection for nearest neighbor classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(06):787–806, 2005.

8 Cenk Baykal, Lucas Liebenwein, Igor Gilitschenski, Dan Feldman, and Daniela Rus. Data-dependent coresets for compressing neural networks with applications to generalization bounds. *arXiv preprint*, 2018. `arXiv:1804.05345`.

9 Jon Louis Bentley and James B Saxe. Decomposable searching problems I. Static-to-dynamic transformation. *Journal of Algorithms*, 1(4):301–358, 1980.

10 Ahmad Biniaz, Sergio Cabello, Paz Carmi, Jean-Lou De Carufel, Anil Maheshwari, Saeed Mehrabi, and Michiel Smid. On the minimum consistent subset problem. In *WADS*, 2019.

11 Vladimir Braverman, Dan Feldman, and Harry Lang. New frameworks for offline and streaming coreset constructions. *arXiv preprint*, 2016. arXiv:1612.00889.

12 Václav Chvatal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 1979.

13 Richard Cole and Lee-Ad Gottlieb. Searching dynamic point sets in spaces with bounded doubling dimension. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 574–583, 2006.

14 Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 1967.

15 Luc Devroye. On the inequality of cover and hart in nearest neighbor discrimination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1:75–78, 1981.

16 Uriel Feige. A threshold of ln n for approximating set cover. *JACM*, 1998.

17 Dan Feldman. Core-sets: Updated survey. In *Sampling Techniques for Supervised or Unsupervised Tasks*, pages 23–44. Springer, 2020.

18 Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578, 2011.

19 Evelyn Fix and Joseph L. Hodges. Discriminatory analysis, nonparametric discrimination: Consistency properties. *US Air Force School of Aviation Medicine*, Technical Report 4(3):477+, January 1951.

20 Alejandro Flores-Velazco and David M. Mount. Guarantees on nearest-neighbor condensation heuristics. In Zachary Friggstad and Jean-Lou De Carufel, editors, *Proceedings of the 31st Canadian Conference on Computational Geometry, CCCG 2019, August 8-10, 2019, University of Alberta, Edmonton, Alberta, Canada*, pages 87–93, 2019.

21 Salvador Garcia, Joaquin Derrac, Jose Cano, and Francisco Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE TPAMI*, 2012.

22 Lee-Ad Gottlieb, Aryeh Kontorovich, and Pinhas Nisnevitch. Near-optimal sample compression for nearest neighbors. In *Advances in Neural Information Processing Systems*, 2014.

23 Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300, 2004.

24 Sariel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006.

25 Peter Hart. The condensed nearest neighbor rule (corresp.). *IEEE Trans. Inf. Theor.*, 1968.

26 Juha Heinonen. *Lectures on analysis on metric spaces*. Springer Science & Business Media, 2012.

27 Norbert Jankowski and Marek Grochowski. Comparison of instances selection algorithms I. Algorithms survey. In *Artificial Intelligence and Soft Computing-ICAISC*. Springer, 2004.

28 Kamyar Khodamoradi, Ramesh Krishnamurti, and Bodhayan Roy. Consistent subset problem with two labels. In *Conference on Algorithms and Discrete Applied Mathematics*, 2018.

29 Lucas Liebenwein, Cenk Baykal, Harry Lang, Dan Feldman, and Daniela Rus. Provable filter pruning for efficient neural networks. *arXiv preprint*, 2019. arXiv:1911.07412.

30 Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM (JACM)*, 41(5):960–981, 1994.

31 David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. Chromatic nearest neighbor searching: A query sensitive approach. *Computational Geometry*, 2000.

32 Azaria Paz and Shlomo Moran. Non deterministic polynomial optimization problems and their approximations. *Theoretical Computer Science*, 15(3):251–277, 1981.

33 Jeff M. Phillips. Coresets and sketches, 2016. arXiv:1601.00617.

34 G. L. Ritter, H. B. Woodruff, S. R. Lowry, and T. L. Isenhour. An algorithm for a selective nearest neighbor decision rule. *IEEE Transactions on Information Theory*, 1975.

35 Petr Slavík. A tight analysis of the greedy algorithm for set cover. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, STOC, 1996.

36 Charles J Stone. Consistent nonparametric regression. *The annals of statistics*, pages 595–620, 1977.

37 Godfried Toussaint. Open problems in geometric methods for instance-based learning. In *JCDCG*, volume 2866 of *Lecture Notes in Computer Science*. Springer, 2002. doi:10.1007/978-3-540-44400-8_29.

38 Murad Tukan, Cenk Baykal, Dan Feldman, and Daniela Rus. On coresets for support vector machines. *arXiv preprint*, 2020. arXiv:2002.06469.

39 Gordon Wilfong. Nearest neighbor problems. In *Proceedings of the Seventh Annual Symposium on Computational Geometry*, SoCG, pages 224–233, New York, NY, USA, 1991. ACM.

40 Anastasiya V. Zukhba. NP-completeness of the problem of prototype selection in the nearest neighbor method. *Pattern Recog. Image Anal.*, 20(4):484–494, 2010.