On Finding Hubs in High Dimensions with Sampling

Huiwen Dong[1], Linghan Zeng[2], Zhiwen Zhao[1], Francesco Silvestri[3], Ninh Pham[2]

[1]Faculty of Artificial Intelligence, Beijing Normal University, Guangdong, China
[2]School of Computer Science, University of Auckland, New Zealand
[3]Department of Information Engineering, University of Padova, Italy
donghw.dhw@gmail.com, lzen402@aucklanduni.ac.nz, mlt.bnu2017@bnu.edu.cn, francesco.silvestri@unipd.it, ninh.pham@auckland.ac.nz

Abstract

Hubs are a few points that frequently appear in the k-nearest neighbors (kNN) of many other points in a high-dimensional data set. The hubs' effects, called the *hubness phenomenon*, degrade the performance of kNN based models in high dimensions. We present *SamHub*, a simple sampling approach to efficiently identify hubs with theoretical guarantees. Apart from previous works based on approximate kNN indexes, SamHub is generic and applicable to any distance measure with negligible additional memory footprint. Empirically, by sampling only 10% of points, SamHub runs significantly faster and offers higher accuracy than existing hub detection methods on many real-world data sets with dot product, L1, L2, and dynamic time warping distances. Our ablation studies of SamHub on improving kNN-based classification show potential for other high-dimensional data analysis tasks.

Introduction

Analyzing high-dimensional data is often challenging due to the presence of hubs, known as the *hubness phenomenon*. Hubs are defined as a few points that frequently occur in the kNN lists of many other points (Radovanović, Nanopoulos, and Ivanović 2010a). This phenomenon has significant implications across various data analysis tasks. In classification, addressing hubness can improve performance in applications such as text-image matching (Liu et al. 2020), recommender system (Hara et al. 2015a), and few-shot learning (Trosten et al. 2023). In clustering, leveraging hub-like centroids as cluster centers has been shown to increase the accuracy of clustering algorithms (Tomašev et al. 2013; Mani and Domeniconi 2020).

While hubness reduction has recently been the subject of much research (Schnitzer et al. 2012; Hara et al. 2015b; Feldbauer and Flexer 2019), hubness verification, i.e. whether hubs exist in a data set, is a prerequisite to any reduction technique. A recent approach for hubness reduction is to transform data sets into a new hubness-resistant space (Trosten et al. 2023). This necessitates the hubness verification in the new space.

Hubness verification and hub identification are important parts of the hubness reduction process but have not received enough attention so far. Unfortunately, identifying hubs by calculating kNN for each point of the data set $\mathcal{X}$ of size n has $O(n^2)$ time, which is infeasible for large-scale data sets. To reduce the cost of executing n kNN queries, approximate nearest neighbor search (ANNS)-based methods (Douze et al. 2024; Malkov and Yashunin 2018) have been utilized (Feldbauer et al. 2018). However, the cost of building ANNS indexes is significant and might dominate the execution time.

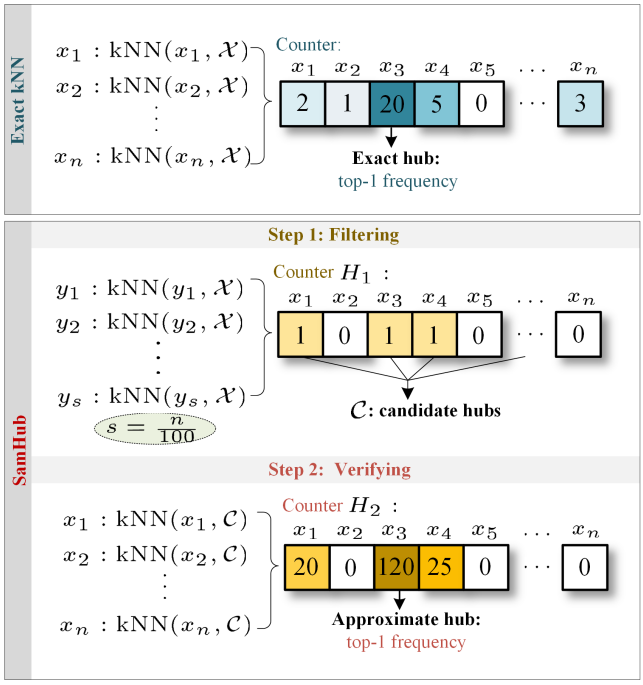

Figure 1: An illustration of SamHub.

Since hubs are the most frequent points on the n kNN lists, given a data set with strong hubness, hubs will likely appear on a small number of sampled kNN lists. Therefore, by considering the set of points on these sampled kNN lists as hub candidates, we can accurately and efficiently identify hubs. Figure 1 briefly shows how our sampling algorithm, called *SamHub*, works. In the exact method, we compute kNN for each $x_i \in \mathcal{X}$ and count the kNN frequency of each point to find the top-1 hub. SamHub has 2 phases, including filtering and verifying steps. In the filtering step, we randomly

sample a subset of s points to find kNN. The union of these sampled kNN will likely contain hubs and hence will be used as the candidate hub $\mathcal{C}$. In the verifying step, the kNN of each point $x_i \in \mathcal{X}$ over the candidate hub $\mathcal{C}$ are sufficient to identify hubs. This is because when the top-1 hub appears in $\mathcal{C}$, its frequency on the verifying step will be significantly higher than the others.

Given $s < n$, SamHub not only breaks the barrier of $O(n^2)$ complexity but also works on arbitrary distance measures. To ensure the reliability of SamHub, we present theoretical guarantees on the accuracy of finding the top-h hubs. Empirically, SamHub runs significantly faster, achieves higher accuracy, and uses fewer memory resources than other ANNS-based approaches for finding hubs on various distance measures, including L1, L2, dot product, and dynamic time warping (Radovanović, Nanopoulos, and Ivanović 2010b). Our ablation studies on utilizing the identified hubs and their *frequency estimate* of being in the kNN lists to improve kNN-based classification show the potential of SamHub for data analysis applications in high-dimensional space.

Related Works

Hubness is a phenomenon related to the curse of dimensionality, stemming from the distance concentration (Radovanović, Nanopoulos, and Ivanović 2010a). A common approach to quantify hubness is to analyze the skewness of the distribution of k-occurrences, the number of times (i.e. frequency) a point appears in the kNN list of other points. Then, the skewness is defined as:

$$S^k = \mathbb{E}\left[\left(O^k - \mu_{O^k}\right)^3\right] / \sigma^3_{O^k}, \tag{1}$$

where μ_{O^k} and σ_{O^k} denote the mean and standard deviation of the k-occurrence distribution O^k, respectively. A higher positive skewness indicates a more severe presence of hubs, as the distribution becomes increasingly right-skewed with a few points occurring frequently as neighbors while most others randomly appear.

Several approaches have been proposed to mitigate the negative effects of hubness. In high-dimensional data analysis, it is essential to repair the asymmetric relationships for accurate similarity measurements. Local scaling (Schnitzer et al. 2012) and non-iterative contextual dissimilarity measure (Jégou, Harzallah, and Schmid 2007) enhance symmetry by adjusting distances according to local densities. Shared nearest neighbors-based measures (Flexer and Schnitzer 2013; Tomašev and Mladenić 2014) enhance symmetry through the intersection of pairwise kNN sets. Flattening methods (Hara et al. 2016) adjust dissimilarity measures to address hubness effects based on the global centroids (Suzuki et al. 2013) or local centroids (Hara et al. 2015b). These solutions require the exact kNN of all data points that run in $O(n^2)$ time for a high-dimensional data set $\mathcal{X}$ of n points.

Recent works have focused on developing fast approximate methods (Feldbauer et al. 2018; Feldbauer and Flexer 2019) that take advantage of industry ANNS libraries like Faiss (Douze et al. 2024), ScaNN (Guo et al. 2020), or Hnswlib (Malkov and Yashunin 2018). Nevertheless, these approaches suffer a significant indexing time and only support popular distance metrics, e.g. L2, dot product.

Algorithm 1: SamHub

Require: Data set $\mathcal{X} \subseteq \mathbb{R}^d$ of n points, s samples, k nearest neighbors, top-h hubs, a distance measure

▷ **Filtering step:**

1: Uniformly sample s points from $\mathcal{X}$ to form $\mathcal{S}$
2: For each $y_i \in \mathcal{S}$, compute the kNN list $kNN(y_i, \mathcal{X})$
3: Form the histogram H_1 where the counter of x_j is the frequency of x_j on the s kNN lists in Line 2
4: Select top-s points with the highest frequency in H_1 to form the candidate set $\mathcal{C}$

▷ **Verifying step:**

5: For each $x_j \in \mathcal{X}$, compute the kNN list $kNN(x_j, \mathcal{C})$
6: Form the histogram H_2 where the counter of x_j is the frequency of x_j in the n kNN lists in Line 5
7: Return top-h points with the highest frequency as top-h hubs and compute the estimated skewness based on H_2

Ensure: The top-h hubs and the estimated skewness score

This work approaches the hubness issue from a new perspective. We estimate the frequency of being in kNN lists for each point (i.e. the histogram output H_2 of SamHub) and utilizing them to improve weighted kNN classifications.

Finding Top-h Hubs with SamHub

We present the intuition of *SamHub* to identify top-h hubs, i.e. the top-h frequent points in n kNN lists of a data set $\mathcal{X}$. We then show the theoretical analysis to guarantee the accuracy of SamHub. For notation, we define $kNN(q, \mathcal{A})$ the list of kNN of q computed *over* the set $\mathcal{A}$ given a distance measure.

We first randomly select s points from $\mathcal{X}$ to form the sample set $\mathcal{S}$. For each $y_i \in \mathcal{S}$, we compute $kNN(y_i, \mathcal{X})$. Since hubs appear in the kNN list of many points in $\mathcal{X}$, the union set $\mathcal{U} = kNN(y_1, \mathcal{X}) \cup \ldots \cup kNN(y_s, \mathcal{X})$ will contain the hubs with high probability.

W.l.o.g., let x_1 be the top-1 hub, and assume that $x_1 \in kNN(x_j, \mathcal{X})$ for a given $x_j \in \mathcal{X}$. If $x_1 \in \mathcal{U} \subseteq \mathcal{X}$, then x_1 must appear on $kNN(x_j, \mathcal{U})$, the kNN list of x_j computed over $\mathcal{U}$. Given $|\mathcal{U}| \leq ks$ (due to duplicates), computing $kNN(x_j, \mathcal{U})$ for all $x_j \in \mathcal{X}$ will break the $O(n^2)$ barrier and identify the hubs with high probability.

While the first *filtering step* computing $kNN(y_i, \mathcal{X})$ for all $y_i \in S$ needs $O(sn)$ distance computations, the second *verifying step* computing $kNN(x_j, \mathcal{U})$ for all $x_j \in \mathcal{X}$ takes $O(ksn)$ distance computations, dominating the running time. To balance the running time of the two steps, we keep the top-s points in $\mathcal{U}$ that have the largest frequency in s $kNN(y_i, \mathcal{X})$ lists as the candidate set $\mathcal{C}$.

Algorithm 1 shows how SamHub works. We first select s points uniformly from $\mathcal{X}$ to form the sample set $\mathcal{S}$, and to construct the frequency histogram H_1. The hub tends to have a high frequency in the histogram H_1 though its frequency might not dominate the non-hub's ones due to sampling. We select the top-s candidate points from H_1 to form $\mathcal{C}$ and compute $kNN(x_i, \mathcal{C})$ for all $x_i \in \mathcal{X}$. $kNN(x_i, \mathcal{C})$ will be used as an approximation of $kNN(x_i, \mathcal{X})$ to identify hubs and estimate the frequency of being in the kNN lists.

Time complexity. SamHub takes $O(sn)$ distance computations, compared to $O(n^2)$ of the exact solution. Since $h < s < n$, the time complexity of constructing the histograms H_1, H_2 is $O(kn + n \log s)$, which is dominated by the distance computational time. We emphasize that SamHub works on arbitrary distance measures, for example, dot product, L1, L2, and dynamic time warping (DTW), with $O(n)$ extra memory footprint to store the histograms.

Theoretical Analysis of SamHub

Let $H_0 = \{f_1, \ldots, f_n\}$ be the *exact* frequency histogram where the counter f_i corresponding to x_i is the k-occurences of x_i, i.e. the number of times x_i in the n $kNN(x_j, \mathcal{X})$ lists for all $x_j \in \mathcal{X}$. We can see that SamHub randomly selects s kNN lists (i.e. $kNN(y_i, \mathcal{X})$ for $y_i \in \mathcal{S}$) to form the histogram $H_1 = \{f'_1, \ldots, f'_n\}$. Since $\mathbf{E}[f'_i] = f_i \cdot s/n$, H_1 can be used to find the hubs if the hubs' frequency is significantly larger than that of non-hubs.

For simplicity, we will show theoretical analysis for $k = 1$. For $k > 1$, we leave it on the supplement. W.l.o.g, we assume $f_1 > f_2 > \ldots > f_n$ when constructing H_0 with $1NN(x_j, \mathcal{X})$ for all $x_j \in \mathcal{X}$. Assume that we are interested in finding a top-1 hub. Hence, x_1 is the top-1 hub with the frequency f_1 and x_2 is the non-hub point with largest frequency f_2 among all non-hubs. Let p_1, p_2 be the probability that x_1, x_2 appear on $1NN(y_i, \mathcal{X})$ for any $y_i \in \mathcal{S}$, respectively, we have $p_1 = f_1/n, p_2 = f_2/n$. The following theorem states the number of samples s required to distinguish between the hub and non-hubs on H_1.

Theorem 1. *Given the hub x_1 with frequency f_1 and the non-hub x_2 with largest frequency f_2, $f_1 > f_2 > 0$. Let $p_1 = f_1/n, p_2 = f_2/n$, and suppose $s \geq \frac{2 \ln n}{(\sqrt{p_1} - \sqrt{p_2})^2}$. With probability at least $1 - 1/n$, the following holds for **all** pairs $x_1, x_j \in \mathcal{X}$: if $f_1 \geq f_j$, then $f'_1 \geq f'_j$.*

Proof. For $i = 1, \ldots, s$, we consider independent pairs of dependent random variables (X_i, Y_i) where

$$X_i = \begin{cases} 1 & \text{if } x_1 = 1NN(y_i, \mathcal{X}); \\ 0 & \text{otherwise.} \end{cases}$$

$$Y_i = \begin{cases} 1 & \text{if } x_2 = 1NN(y_i, \mathcal{X}); \\ 0 & \text{otherwise.} \end{cases}$$

Define $X = \sum_{i=1}^{s} X_i, Y = \sum_{i=1}^{s} Y_i$ the frequency of the hub x_1 and non-hub x_2 on H_1. We consider the failure case $Y > X$ where x_2 is ranked higher than x_1 on H_1. Applying Markov inequality for any $\lambda > 0$, we have

$$\mathbf{Pr}[Y - X > 0] = \mathbf{Pr}\left[e^{\lambda(Y-X)} > 1\right] \leq \mathbf{E}\left[e^{\lambda(Y-X)}\right]$$
$$= \mathbf{E}\left[e^{\lambda(\sum_i Y_i - \sum_i X_i)}\right] = \prod_{i=1}^{s} \mathbf{E}\left[e^{\lambda(Y_i - X_i)}\right].$$

Since $X_i = 1$ with probability $p_1 = f_1/n$, and $Y_i = 1$ with probability $p_2 = f_2/n$, we have

$$\mathbf{E}\left[e^{\lambda(Y_i - X_i)}\right] = e^{\lambda} p_2 + e^{-\lambda} p_1 + (1 - p_1 - p_2)$$
$$\geq 2\sqrt{p_1 p_2} + 1 - p_1 - p_2 = 1 - \left(\sqrt{p_1} - \sqrt{p_2}\right)^2.$$

The equality holds when $\lambda = \ln \sqrt{p_1/p_2} > 0$. By choosing $\lambda = \ln \sqrt{p_1/p_2}$, $\mathbf{Pr}[Y - X > 0]$ is upper bounded by

$$\left(1 - \left(\sqrt{p_1} - \sqrt{p_2}\right)^2\right)^s \leq e^{-s\left(\sqrt{p_1} - \sqrt{p_2}\right)^2}.$$

By choosing $s \geq \frac{2 \ln n}{(\sqrt{p_1} - \sqrt{p_2})^2}$, we have $\mathbf{Pr}[Y - X > 0] \leq 1/n^2$. By the union bound, the claim holds with probability at least $1 - 1/n$. □

Remark. Since SamHub selects top-s points with the highest frequency in H_1, Theorem 1 shows that if $\sqrt{p_1} - \sqrt{p_s} \geq \sqrt{2 \ln(n)/s}$, then $x_1 \in \mathcal{C}$ with probability $1 - 1/n$. While p_1 is determined by the k-occurrences of the hub, $p_s \to 1/n$ for a non-hub x_s. Therefore, selecting $s = 2 \ln(n)/p_1$ guarantees to the top-1 hub in $\mathcal{C}$ with high probability. In addition, the higher k-occurrences the hub is, the smaller s and faster algorithm we have.

Since we can guarantee that the top-1 hub $x_1 \in \mathcal{C}$, and since $\mathcal{C} \subseteq \mathcal{X}$, the counter of H_2 associated with x_1 will be significantly higher than the other non-hubs. This comes from the practical observation that non-hubs tend to uniformly appear in $kNN(x_j, \mathcal{C})$ for each $x_j \in \mathcal{X}$.

Rigorous guarantees on the output. Removing the assumption of uniformly distribution of non-hubs in $kNN(x_j, \mathcal{C})$ requires a careful design. We first assume the frequency f_0 of the true hub x_0, and will later remove this requirement. To provide rigorous guarantees, we replace the verifying step (Lines 5 – 7) by a *validating step* that executes another sampling as follows:

1. Uniformly sample s' points $\mathcal{S}'$ from $\mathcal{X}$.
2. For each $y_i \in \mathcal{S}'$, compute $1NN(y_i, \mathcal{X})$.
3. Form the histogram H'_2 where the counter of x_j is the frequency of x_j in the s' 1NN lists in Step 2.
4. Return the set $\mathcal{C}'$ of candidate hubs that contains all points in $\mathcal{C}$ whose frequency in H'_2 is above $s'(f_0/n)(1 - \epsilon/2)$.

We claim the following result:

Theorem 2. *Let f_0 be the frequency of the true hub x_0, and let $\epsilon \in (0, 1)$. If $s' = \Theta\left(\frac{n}{f_0 \epsilon^2} \log n\right)$, with probability at least $1 - 1/n$, the validating step returns a set $\mathcal{C}'$ that satisfies the following properties:*

- *The hub point x_0 is in $\mathcal{C}'$;*
- *Each point in $\mathcal{C}'$ has frequency at least $(1 - \epsilon) f_0$.*

Proof. We recall the Chernoff bounds for a binomial random variable X with $\mathbf{E}[X] = \mu$. Given $\delta \in (0, 1)$, we have

- (Upper tail) $\mathbf{Pr}[X \geq (1 + \delta)\mu] \leq e^{-\frac{\delta^2 \mu}{3}}$.
- (Lower tail) $\mathbf{Pr}[X \leq (1 - \delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2}}$.

We observe that the hub x_0 is not in the final output $\mathcal{C}'$ if one of the two following events happens:

- The hub x_0 is not in H'_2: This happens with probability $a_0 = (1 - f_0/n)^{s'} \leq e^{-s' f_0/n}$. By setting $s' = \Theta\left(\frac{n}{f_0 \epsilon^2} \log n\right)$, we get $a_0 \leq 1/(4n)$.

	ADL1	Arr	CNAE	Dexter	Diabetes	DNA	Gisette	Heart	Isolet	Mini-ngs	Mutants	Optdigits	Splice	CBF	Lightning2
n	51116	452	1080	300	768	1186	6000	270	7796	1353	31420	5619	1000	899	61
d	250	279	856	20000	9	181	5000	14	616	27226	5408	63	61	128	638
# clu.	5	13	9	2	2	3	2	2	26	4	8	10	2	3	2

Table 1: Data sets: CBF and Lightning2 are time-series while the rest are in high dimensions.

- The hub x_0 is in $\mathcal{C}$, but not in $\mathcal{C}'$: This happens with probability a_1 when the frequency of x_0 in H_2' is less than $s'(f_0/n)(1-\epsilon/2)$. Let X be the binomial random variable denoting the frequency of x_0 in H_2', and let $\mu = s'f_0/n$ be its expectation. By using a Chernoff bound for the lower tail, we get:

$$a_1 = \mathbf{Pr}\left[X \leq \frac{s'f_0}{n}\left(1-\frac{\epsilon}{2}\right)\right] \leq e^{-\frac{s'f_0}{2n}\frac{\epsilon^2}{4}}.$$

Given $s' = \Theta\left(\frac{n}{f_0\epsilon^2}\log n\right)$, we get $a_1 \leq 1/(4n)$.

By the union bound, we have $\mathbf{Pr}\left[x_0 \notin \mathcal{C}'\right] \leq 1/(2n)$.

We now upper bound the probability that any point x_i with $f_i < (1-\epsilon)f_0$ is returned in $\mathcal{C}'$. If $x_i \in \mathcal{C}$, then $x_i \in \mathcal{C}'$ when its frequency in H_2' above $s'(f_0/n)(1-\epsilon/2)$. Let Y be the binomial random variable denoting the frequency of x_i in H_2'. It is clear that $\mathbf{E}\left[Y\right] = s'f_i/n \leq s'(1-\epsilon)f_0/n$. By using a Chernoff bound for the upper tail, we get:

$$\begin{aligned} a_2 &= \mathbf{Pr}\left[Y \geq \frac{s'f_0}{n}\left(1-\frac{\epsilon}{2}\right)\right] \\ &= \mathbf{Pr}\left[Y \geq \frac{(1-\epsilon)s'f_0}{n}\left(1+\frac{\epsilon}{2(1-\epsilon)}\right)\right] \\ &\leq e^{-\frac{(1-\epsilon)s'f_0}{n}\frac{\epsilon^2}{12(1-\epsilon)^2}} \\ &\leq e^{-\frac{s'f_0}{n}\frac{\epsilon^2}{12(1-\epsilon)}} \leq e^{-\frac{s'f_0}{n}\frac{\epsilon^2}{12}}. \end{aligned}$$

By setting $s' = \Theta\left(\frac{n}{f_0\epsilon^2}\log n\right)$, we get $a_2 \leq 1/(2n^2)$. Then, by an union bound, the probability of returning a point x_i with frequency lower than $(1-\epsilon)f_0$ is at most $1/2n$. By putting together the probability of not returning a hub or return a non-frequent point, we get that the failure probability is at most $1/n$ and the theorem follows. □

The validating step requires to know the frequency of the true hub f_0. We can remove this requirement by using the standard doubling trick. Specifically, we initially run the algorithm with $f_0 = n$ and $f_0 = n/2$, obtaining a candidate sets $\mathcal{C}_n'$ and $\mathcal{C}_{n/2}'$, respectively. If $\mathcal{C}_{n/2}'$ contains a point of $\mathcal{C}_n'$, we stop and return $\mathcal{C}_n'$. Otherwise, we repeat the following procedure until it stops or $i = \log n$. Assume that $\mathcal{C}_{n/2^i}'$ is already known, we compute $\mathcal{C}_{n/2^{i+1}}'$ by running the algorithm with $f_0 = n/2^{i+1}$. If the new set contains a point from $\mathcal{C}_{n/2^i}'$, then we return $\mathcal{C}_{n/2^i}'$; otherwise, we continue with $i = i+1$.

While we can provide rigorous guarantees on SamHub with the validating step, we runs SamHub with the verifying step in practice. This is because non-hubs tend to uniformly appear in $kNN(x_j, \mathcal{C})$ for each $x_j \in \mathcal{X}$. Importantly, the histogram H_2 provided by the verifying step will be used to estimate the skewness score and the frequency of each point to improve the weighted kNN classification.

Experiments

We implement *SamHub* with a few lines of Python codes[1]. We conduct experiments on a 3.2 GHz Core i7-5800 CPU with 64GB of RAM. We present empirical evaluations on the accuracy of hub detection to verify our claims, including:

- SamHub provides high utility in estimating the skewness score and accurately identifying the top-h hubs closely aligning with the results from the exact solution.
- SamHub works across non-metric and metric distances, outperforming ANNS-based methods in terms of efficiency and accuracy.
- The frequency estimation provided by SamHub leads to an improvement of a weighted kNN classification.

Our competitors are hub detectors using ANNS indexes including Faiss, Hnswlib, and ScaNN with optimal settings for each data set. Details of these settings are in the supplement. In contrast, SamHub does not need hyper-parameters except for the sample size s. We conduct experiments on many real-world data sets in Table 1. Details of the characteristics of the data sets are in the supplement. All results are the average of 5 runs of the algorithms.

We measure the accuracy of top-h hub detectors as follows.

$$\texttt{Accuracy} = \frac{|\text{Exact top-}h\text{ hubs} \cap \text{Detected top-}h\text{ hubs}|}{h}.$$

We report the total execution time of SamHub to identify top-h hubs. For ANNS-based methods, we report separately the indexing and querying time to show the significance of indexing. For all experiments, we set $h = 10$ and vary k and s. Empirical results on various h are in the supplement.

Accuracy of SamHub on Finding Top-10 Hubs

This subsection shows the accuracy of SamHub on detecting top-10 hubs on high-dimensional data sets with L1, L2, and dot product distances, and time series data sets with dynamic time warping (DTW), given $k = 10$. Figure 2 shows the accuracy of SamHub on many real-world data sets with various skewness while varying the number of samples $s = \{2\%, \ldots, 20\%\} \cdot n$.

It is clear that SamHub increases the accuracy of identifying hubs when increasing the sample size s. The accuracy of SamHub also depends on the skewness score of each data

[1] https://github.com/AnnieDong-code/SamHub.git

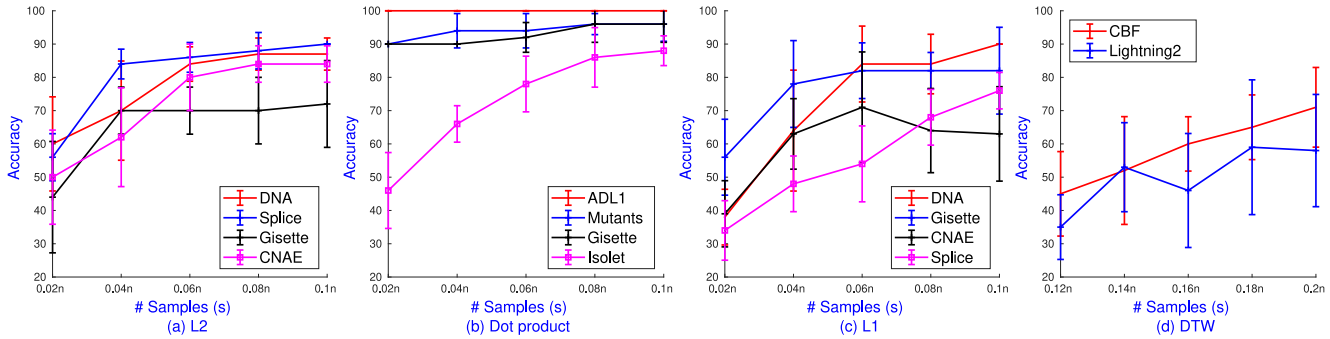

Figure 2: The accuracy of SamHub over a wide range of s on L2, dot product, L1, and DTW distances ($h = k = 10$).

Data sets	Exact	SamHub		Hnswlib			Faiss-IVFPQ			ScaNN		
	Time	Time	Acc	Index	Query	Acc	Index	Query	Acc	Index	Query	Acc
Gisette (L2)	3.63s	**0.85s**	**70%**	1.32s	1.17s	60%	2.38s	1.29s	20%	-	-	-
Mutants (Dot)	114.77s	**2.30s**	**90%**	2.71s	2.34s	**90%**	6.78s	1.92s	0%	82.64s	15.35s	**90%**
ADL1 (Dot)	34.78s	**0.25s**	**100%**	0.30s	0.34s	**100%**	0.43s	1.71s	0%	5.78s	1.90s	**100%**

Table 2: Comparison of running time and accuracy between SamHub, exact and ANNS-based approaches ($h = k = 10$).

Data sets	Exact	Hnswlib	Faiss	ScaNN
Gisette (L2)	4.30 ×	2.93 ×	4.32 ×	-
Mutants (Dot)	49.90 ×	2.20 ×	3.78 ×	42.60 ×
ADL1 (Dot)	139.1 ×	2.56 ×	8.56 ×	30.70 ×

Table 3: Speed up of SamHub against exact and ANNS-based approaches ($h = k = 10$).

set and the used distance measures. As the skewness of the used data sets with dot product is highest, SamHub gives the highest accuracy given the same sampling ratio. On the other hand, the skewness of 2 time series data is lower, leading to larger sampling ratios to achieve reasonable accuracy.

Figure 2(a) displays results using L2 on the Splice, DNA, Gisette, and CNAE data sets with skewness of 4.17, 4.21, 4.66, and 5.90, respectively. It shows that as the sampling ratio increases, SamHub's accuracy improves, reaching above 60% accuracy at a 4% ratio for all data sets, and 85% accuracy at a 10% ratio for all studied data sets.

In Figure 2(b), results are presented for the dot product. We analyze Isolet, Gisette, Mutants, and ADL1 data sets with skewness of 7.86, 19.71, 30.27, and 70.88, respectively. SamHub achieves a stable 100% accuracy on the highly skewed ADL1 data set at only 2% sampling ratio, while the accuracy for Mutants and Gisette achieves around 90%. For Isolet with relatively lower skewness, the accuracy increases steadily and up to 80% at a 10% sampling ratio.

Figure 2(c) illustrates the results using L1 on Splice, DNA, Gisette, and CNAE, with skewness of 2.32, 4.08, 5.75, and 7.05, respectively. The accuracy shows a consistent improvement with an increasing sampling ratio, achieving over 60% accuracy at an 8% sampling ratio and exceeding 75% for Splice, DNA, and Gisette at a 10% sampling ratio.

Finally, Figure 2(d) shows results for DTW on two time series data sets, including CBF with a skewness of 2.49 and Lightning2 with a skewness of 0.63. The accuracy of SamHub improves steadily as the sampling ratio increases. For CBF, the accuracy rises significantly from 45% to 70% as the sampling ratio increases from 12% to 20%.

The results demonstrate the effectiveness of SamHub for metric and non-metric distances across data sets with different levels of skewness. For lower skewness data sets (e.g. time series data sets with DTW), we need a larger s to increase the gap between p_1 and p_s to gain a higher accuracy, as explained in our theoretical results.

Compared with ANNS-based Approaches

This subsection compares the efficiency of SamHub with ANNS-based methods on L2 and dot product since they do not support L1 and DTW. Since ANNS libraries have two running time components, including indexing and querying, we carefully tune their parameters to achieve a high accuracy with the smallest total execution time. These parameter settings highly depend on the data sets and the distance measures, and we leave them to the supplement. For SamHub, we select the smallest s to reach a similar accuracy as ANNS-based methods. Note that ScaNN does not support L2.

Table 2 shows the running time and accuracy of SamHub, the exact and ANNS-based solutions, including Hnswlib, Faiss-IVFPQ, and ScaNN on Gisette with L2, and Mutants and ADL1 on dot product. It is clear that the running time of SamHub is similar to the indexing time of these ANNS methods. For ANNS-based hub detection methods, we separate indexing and querying time though their total execution time includes both components.

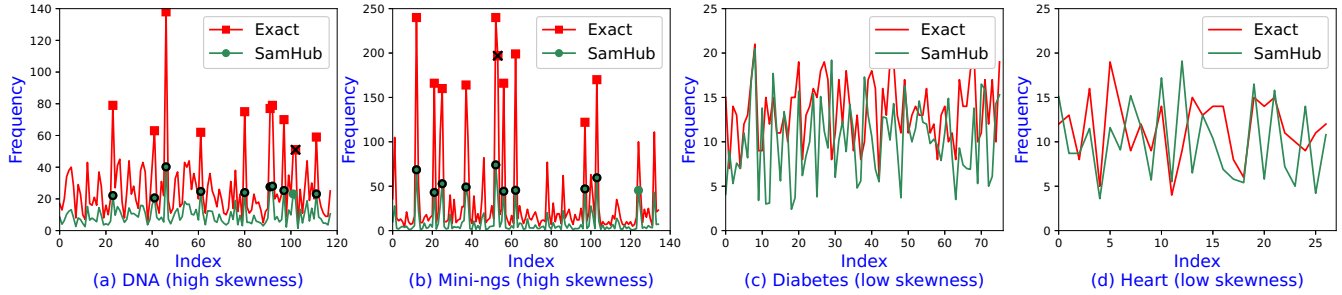

Figure 3: Comparison between the histogram H_2 to estimate k-occurrences ($h = k = 10, s = 0.1n$) and the exact solution.

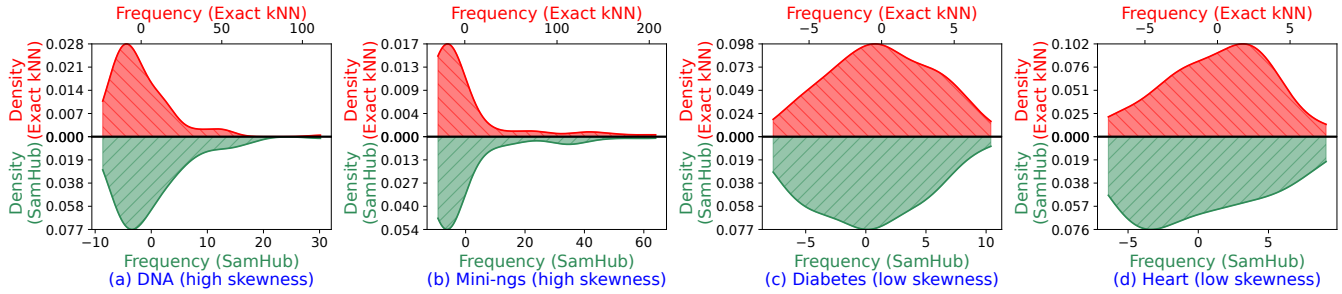

Figure 4: Comparison of pdf of k-occurrences provided by H_2 ($h = k = 10, s = 0.1n$) and the exact solutions.

On L2, SamHub achieves an accuracy of 70% with a total running time of 0.85 seconds on Gisette. SamHub is 10% more accurate than Hnswlib while being 1.64s faster. Additionally, SamHub is significantly more accurate than Faiss-IVFPQ while also speeding up the running time by 2.82s.

On dot product, SamHub, Hnswlib, and ScaNN achieve a high accuracy of 90% on Mutants. However, SamHub is the fastest, with a total running time of 2.3 seconds, speeding up against the exact method by 112.47s, Hnswlib by 2.75s, Faiss-IVFPQ by 6.40s, and ScaNN by 95.69s. On ADL1, SamHub achieves a perfect accuracy of 100% in just 0.25s, which is faster than exact, Hnswlib, Faiss-IVFPQ, and ScaNN by 34.53s, 0.39s, 1.89s, and 7.43s, respectively.

We summarize the speed up of SamHub against those competitors in Table 3. These results demonstrate the competitive performance of SamHub compared to ANNS-based approaches in both accuracy and running time. While SamHub only requires s to govern the tradeoff between accuracy and efficiency, ANNS-based solutions need to tune several parameters for both indexing and querying steps.

The Utility of Estimated k-occurrences by SamHub

This subsection verifies the utility of SamHub on both high and low skewness data sets in estimating k-occurrences and identifying top-h hubs, given $k = 10$. We use the histogram H_2 provided by SamHub to estimate the k-occurrences, and compare it with the exact solution.

Figure 3 shows the estimated k-occurrences of SamHub and the exact method in both high and low skewness data sets. Since we use $s = 0.1n$, we scale down 10 times the frequency provided by SamHub to give an appropriate scale compared to the exact solution. Such scaling does not affect the order and the relative gap of the estimated k-occurrences of SamHub. To facilitate comparison, we remove the data with k-occurrences of zero in SamHub, and hence remove these points from the exact results.

Figure 3(a) and (b) present the results for high skewness data sets, i.e. DNA with skewness of 4.21 and Mini-ngs with skewness of 8.45 on L2. It is clear that SamHub estimates k-occurrences closely align with those by exact methods and identifies the hubs with high accuracy. Using the first 10 points with the highest k-occurrences as hubs, the top-10 hubs identified by SamHub are marked with green circles, and those by exact kNN are marked with red squares. The 9 identified hubs are marked with black circles, while the 1 missed by SamHub is indicated with a black cross.

Figure 3(c) and (d) show the results for low skewness data sets, i.e. Diabetes with skewness of 0.17 and Heart with skewness of 0.02 on L2. Conversely, low skewness in exact kNN indicates a more uniform distribution with no prominent hubs. Similarly, SamHub exhibits a uniform k-occurrence distribution, indicating the absence of significantly skewed hubs and the low effects of hubness phenomenon.

This utility is further illustrated as the probability density function (pdf) in Figure 4 of these 4 data sets where the upper part shows the pdf of exact kNN, and the lower part represents SamHub. Both the exact kNN's and SamHub's means are subtracted for better visualization. It is easy to see long tails of the distribution of k-occurrences on high skewness data sets in Figure 4(a) and (b), which indicate the strong effect of hubness phenomenon.

On low skewness data sets in Figure 4(c) and (d), there

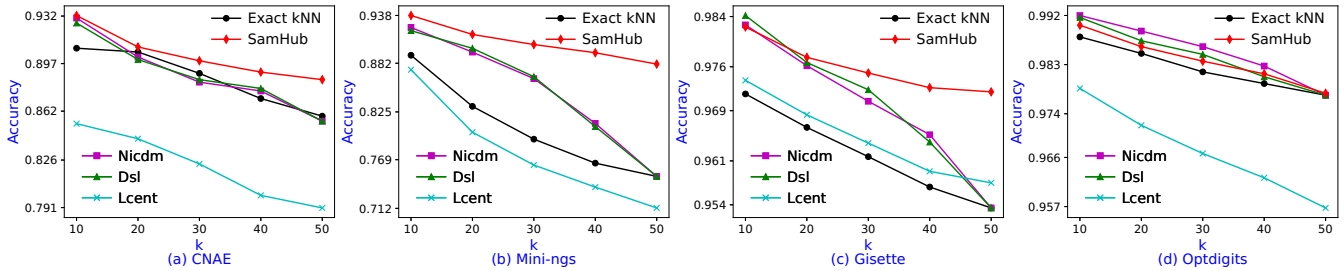

Figure 5: Accuracy of kNN classifiers: A weighted kNN with the estimated k-occurrences provided by SamHub ($h = 10, s = 0.1n$), unweighted kNN, and other popular hubness reduction methods.

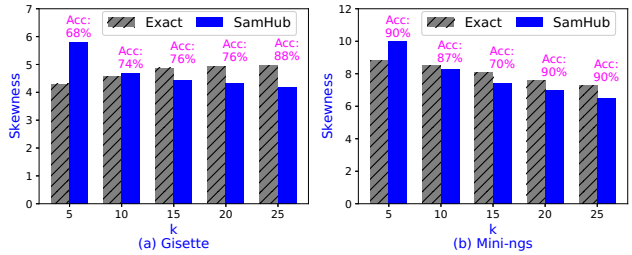

Figure 6: The estimated skewness S^k and top-10 hubs accuracy for different k of SamHub ($s = 0.1n$).

are no long tails in the k-occurrence distribution. Also, these distributions towards the right are more uniform, indicating the absence of skewed hubs. These results demonstrate the utility of SamHub as it is consistent with the exact kNN in estimating k-occurrences and identifying top-h hubs across different skewness levels of the data.

The Ablation Study in kNN Classification

This subsection shows the ablation study on kNN classifiers to explore the impact of hubs identified by SamHub. As hubs negatively affect kNN classifiers by providing incorrect classification information to other points, implementing a simple removal or a weighted kNN scheme to penalize identified hubs is expected to improve the classification performance.

To test this hypothesis, we implement a weighted kNN classification where the weight of a point is derived by its estimated k-occurrence score by SamHub. Similar to Radovanović, Nanopoulos, and Ivanović (2010a), the weight of x_i is as follows:

$$w(x_i) = \exp\left(-\frac{f_i - \mu}{\sigma}\right),$$

where f_i is the estimated k-occurrences of x_i provided by H_2 of SamHub, and $\mu = \frac{1}{n}\sum_{i=1}^{n} f_i, \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - \mu)^2}$.

Figure 5 compares the classification accuracy between the exact kNN, hubness reduction methods including Nicdm (Jégou, Harzallah, and Schmid 2007), Dsl (Hara et al. 2016), and Lcent (Hara et al. 2015b), and the SamHub-based weighted method across different values of k. The CNAE, Mini-ngs, Gisette, and Optdigits data sets exhibit skewness of 5.90, 8.45, 4.66, and 1.02 on L2. Obviously, the SamHub-based method shows a consistent improvement over the competitors. SamHub-based weighted kNN gives a stable accuracy over a wide range of k while the performance of other competitors are very sensitive with k. For example, on Figure 5(b), it improves the accuracy of the exact kNN, Nicdm, Dsl, and Lcent by at least 10% with $k = 50$.

These results demonstrate that leveraging the frequency estimate by SamHub can effectively enhance the kNN classification accuracy across various data sets. The effectiveness of SamHub for the classification indicates the potential for similar downstream tasks in high-dimensional space.

The Sensitivity of k

This subsection verifies the utility of SamHub in estimating skewness and identifying top-h hubs across different k. Figure 6 displays SamHub results on Gisette with a skewness of 4.66 and Mini-ngs with a skewness of 8.45 using L2 while varying k.

It is clear that the skewness by SamHub remains closely aligned with the exact kNN with different k on both Gisette and Mini-ngs. The skewness of SamHub slightly decreases when increasing k because the exact k-occurrences of top-h hubs slightly decrease the magnitude. Nevertheless, the accuracy of the top-10 hubs is still remains reliable, ranging between 70% and 90% as k increases, as shown in the red text in the figure.

Conclusion

The paper proposes SamHub, a simple sampling approach to efficiently identify hubs with theoretical guarantees. SamHub runs in linear time, requiring $O(sn)$ distance computations, with negligible memory footprint. Theoretically, SamHub can govern the time-accuracy tradeoff on identifying hubs via the sample size s. Empirically, SamHub runs significantly faster, achieves higher accuracy, and uses less memory footprint than other ANNS-based approaches for finding hubs on various distance measures, including L1, L2, dot product, and dynamic time warping.

Our ablation study on kNN classification that leverages the frequency estimation provided by SamHub shows potential on improving other data analysis in high-dimensional space. We hope that the use of estimated frequency by SamHub will be further explored in clustering and outlier detection tasks.

Acknowledgments

This work was done at University of Auckland during Dong's visit with the support of the China Scholarship Council (CSC) program (Project ID: 202306040077). Zeng, Silvestri and Pham are supported by Marsden Fund (MFP-UOA2226). Silvestri is also supported by the Uni-Impresa (Big-Mobility) and MUR PRIN (20174LF3T8 *AHeAD*) funds.

References

Douze, M.; Guzhva, A.; Deng, C.; Johnson, J.; Szilvasy, G.; Mazaré, P.-E.; Lomeli, M.; Hosseini, L.; and Jégou, H. 2024. The Faiss Library.

Feldbauer, R.; and Flexer, A. 2019. A Comprehensive Empirical Comparison of Hubness Reduction in High-Dimensional Spaces. *Knowledge and Information Systems*, 59(1): 137–166.

Feldbauer, R.; Leodolter, M.; Plant, C.; and Flexer, A. 2018. Fast Approximate Hubness Reduction for Large High-dimensional Data. In *Proceedings of the IEEE International Conference on Big Knowledge (ICBK)*, 358–367. IEEE.

Flexer, A.; and Schnitzer, D. 2013. Can Shared Nearest Neighbors Reduce Hubness in High-dimensional Spaces? In *Proceedings of the IEEE 13th International Conference on Data Mining Workshops*, 460–467. IEEE.

Guo, R.; Sun, P.; Lindgren, E.; Geng, Q.; Simcha, D.; Chern, F.; and Kumar, S. 2020. Accelerating Large-scale Inference with Anisotropic Vector Quantization. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR.

Hara, K.; Suzuki, I.; Kobayashi, K.; and Fukumizu, K. 2015a. Reducing Hubness: A Cause of Vulnerability in Recommender Systems. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 815–818. ACM.

Hara, K.; Suzuki, I.; Kobayashi, K.; Fukumizu, K.; and Radovanović, M. 2016. Flattening the Density Gradient for Eliminating Spatial Centrality to Reduce Hubness. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 1659–1665. AAAI.

Hara, K.; Suzuki, I.; Shimbo, M.; Kobayashi, K.; Fukumizu, K.; and Radovanović, M. 2015b. Localized Centering: Reducing Hubness in Large-sample Data. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2645–2651. AAAI.

Jégou, H.; Harzallah, H.; and Schmid, C. 2007. A Contextual Dissimilarity Measure for Accurate and Efficient Image Search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. IEEE.

Liu, F.; Ye, R.; Wang, X.; and Li, S. 2020. HAL: Improved Text-Image Matching by Mitigating Visual Semantic Hubs. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 11563–11571. AAAI.

Malkov, Y.; and Yashunin, D. 2018. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4): 824–836.

Mani, P.; and Domeniconi, C. 2020. Hub-based Subspace Clustering. *Neurocomputing*, 413: 193–209.

Radovanović, M.; Nanopoulos, A.; and Ivanović, M. 2010a. Hubs in Space: Popular Nearest Neighbors in High-dimensional Data. *Journal of Machine Learning Research*, 11(Sept): 2487–2531.

Radovanović, M.; Nanopoulos, A.; and Ivanović, M. 2010b. Time-series Classification in Many Intrinsic Dimensions. In *Proceedings of the SIAM International Conference on Data Mining (SDM 2010)*, 677–688. SIAM.

Schnitzer, D.; Flexer, A.; Schedl, M.; and Widmer, G. 2012. Local and Global Scaling Reduce Hubs in Space. *Journal of Machine Learning Research*, 13(Oct): 2871–2902.

Suzuki, I.; Hara, K.; Shimbo, M.; Saerens, M.; and Fukumizu, K. 2013. Centering Similarity Measures to Reduce Hubs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 613–623. ACL.

Tomašev, N.; and Mladenić, D. 2014. Hubness-aware Shared Neighbor Distances for High-dimensional k-Nearest Neighbor Classification. *Knowledge and Information Systems*, 39(1): 89–122.

Tomašev, N.; Radovanović, M.; Mladenić, D.; and Ivanović, M. 2013. The Role of Hubness in Clustering High-dimensional Data. *IEEE Transactions on Knowledge and Data Engineering*, 26(3): 739–751.

Trosten, D. J.; Chakraborty, R.; Løkse, S.; Jenssen, R.; and Vidal, R. 2023. Hubs and Hyperspheres: Reducing Hubness and Improving Transductive Few-shot Learning with Hyperspherical Embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7527–7536. IEEE.