HubHSP Graph: Effective Data Sampling for Pivot-Based Representation Strategies

Stephane Marchand-Maillet[1(✉)] and Edgar Chávez[2]

[1] Department of Computer Science, University of Geneva, Geneva, Switzerland
stephane.marchand-maillet@unige.ch
[2] CICESE, Ensenada, Mexico

Abstract. Given a finite dataset in a metric space, we investigate the definition of a representative sample. Such a definition is important in data analysis strategies to seed algorithms (such as k-means) and for pivot-based data indexing techniques. We discuss the geometrical and statistical facets of such a definition.

We propose the Hubness Half Space Partitioning (HubHSP) strategy as an effective sampling heuristic that combines both geometric and statistical constraints. We show that the HubHSP sampling strategy is sound and stable in non-uniform high-dimensional regimes and compares favorably with classical sampling techniques.

Keywords: Dataset sampling · Pivot-based indexing · Local intrinsic dimensionality · Hubness half space partitioning

1 Introduction

Given a dataset in a metric space, the selection of a *representative subset* of the dataset is a common operation in data analysis or for data indexing. It is well known that obtaining a decent approximation of cluster centers prior to running a clustering algorithm such as k-means improves not only the speed of convergence but also the quality of the final result [2].

Pivot-based exact and approximate indexing techniques are based on the prior selection of a *pivot set* which is used in two main mechanisms. Defining pivots as landmarks in the metric space allows to precompute and store distance values from all data to this set and use this information along with the triangle inequality to build an *exclusion criterion* [5].

Pivots may also be used as landmarks to represent the data in permutation-based indexing strategies. The query locates data in its neighborhood by activating pivots and selecting data with similar activation. In both cases the idea is to restrict the number of data for which the exact distance computation is performed [1,3,10]

In the parallel field of data visualization of large data (outside the scope of this paper) the smart sub-sampling of the dataset into a reduced representative subset ensures smooth and accurate display.

T. Skopal et al. (Eds.): SISAP 2022, LNCS 13590, pp. 164–177, 2022.
https://doi.org/10.1007/978-3-031-17849-8_13

In this paper, we first study the approaches for data sampling and the possible constraints that can be set, namely statistical or geometrical. We then propose the Hubness Half Space Partitioning (HubHSP) that builds on the Half Space Partitioning (HSP [4]) to construct a data selector that effectively combines such geometrical and statistical constraints.

We demonstrate empirically the validity and stability of our proposal in various experimental conditions.

2 Dataset Sampling Strategies

Given a N-sized dataset $\mathcal{X} = \{x_i\}_{i\in[\![N]\!]}$ of $\Omega \subseteq \mathbb{R}^D$, classical data sampling strategies are generally either based on statistical or geometric constraints.

2.1 Density-Based Sampling

One natural way to approach dataset re-sampling is from a statistical perspective. Here, the dataset $\mathcal{X}$ is supposed to be a N-sized i.i.d sample of a probability density function (pdf) $f_{\mathcal{X}}$. In other words, $\{x_i\}_{i\in[\![N]\!]}$ is one realization of a set of N independent random variables $\{X_i\}_{i\in[\![N]\!]}$ identically distributed according to this pdf ($X_i \sim f_{\mathcal{X}}$, $i \in [\![N]\!]$).

Re-sampling dataset $\mathcal{X}$ into subset $\mathcal{Y} = \{y_j\}_{j\in[\![n]\!]}$ with $n \leq N$ therefore amounts to make a selection $\mathcal{Y} \subseteq \mathcal{X}$ of n data from $\mathcal{X}$ into $\mathcal{Y}$. In this case a subset of indices $i_j \in [\![N]\!]$ is chosen so that $y_j = x_{i_j} \forall j \in [\![n]\!]$. As shown below, a uniform sampling of indices from within $[\![N]\!]$ guarantees that $\mathcal{Y}$ is also a sample of pdf $f_{\mathcal{X}}$ (i.e. $f_{\mathcal{Y}} = f_{\mathcal{X}}$).

Representation Properties. Maintaining the probability density function of a sample has specific implications. Statistically, a high value of the pdf at a location $x \in \Omega$ makes the likelihood of a sample at this location $\mathrm{P}(X_i = x)$ accordingly high.

Conversely, a crude empirical estimate of the value of the pdf at location x, $\hat{f}_{\mathcal{X}}(x)$ is given by the density of samples from $\mathcal{X}$ around x. Classically, the density is defined as the number of objects of interest per unit of volume. Hence, we can define

$$\hat{f}_{\mathcal{X}}(x) = \frac{|\mathcal{X} \cap \mathcal{B}(x,\rho)|}{\mathrm{vol}(\mathcal{B}(x,\rho))} \quad \text{for some small } \rho > 0$$

where we consider the ball $\mathcal{B}(x,\rho) = \{y \in \Omega \mid d(x,y) \leq \rho\}$ as a unit volume. In practice, we only have access to the data from $\mathcal{X}$. Hence the estimate is only non-zero when the ball $\mathcal{B}(x,\rho))$ contains data samples. As a result, we are led to using the k nearest neighbors of x from $\mathcal{X}$ ($\mathcal{V}^k_{\mathcal{X}}(x)$) to estimate the density:

$$\hat{f}_{\mathcal{X}}(x) = \frac{k}{\mathrm{vol}(\mathcal{V}^k_{\mathcal{X}}(x))} \quad \text{for some } k > 0 \tag{1}$$

Note that following the above, the volume $\mathrm{vol}(\mathcal{V}^k_{\mathcal{X}}(x))$ can be the volume of the enclosing ball ($\mathrm{vol}(\mathcal{V}^k_{\mathcal{X}}(x)) = \mathrm{vol}(\mathcal{B}(x,\rho))$ with ρ the distance to the k^{th} neighbor).
This view justifies that $\hat{f}_{\mathcal{X}}(x) = \hat{f}_{\mathcal{Y}}(x)$ as follows [11]:

Let $\mathrm{P}(x_j \in \mathcal{V}^k_{\mathcal{X}}(x_i)) = p_{j|i}$

then $\mathrm{P}(x_j \in \mathcal{V}^k_{\mathcal{Y}}(x_i)) = \mathrm{P}(x_j \in \mathcal{V}^k_{\mathcal{X}}(x_i), x_j \in \mathcal{Y}) \overset{\perp\!\!\!\perp}{=} p_{j|i}\,\mathrm{P}(x_j \in \mathcal{Y})$.

If we sample uniformly n indices $j \in [\![N]\!]$ then $\mathrm{P}(x_j \in \mathcal{Y}) = \frac{n}{N}$. As a result, $\mathrm{P}(x_j \in \mathcal{V}^k_{\mathcal{Y}}(x_i)) \propto p_{j|i}$ and the normalization ensures that $\hat{f}_{\mathcal{X}}$ and $\hat{f}_{\mathcal{Y}}$ are estimates of the same original density $f_{\mathcal{X}}$. □

This also pinpoints the fact that since $\mathcal{Y} \subseteq \mathcal{X}$ preserves the original density $f_{\mathcal{X}}$ then $\mathcal{X}$ can be uniformly partitioned into equivalence classes whose representative centers are points $x_j \in \mathcal{Y}$ and the respective radii depend on the local density.

From (1), for a fixed k, $\hat{f}_{\mathcal{X}}$ varies according to the value of $\mathrm{vol}(\mathcal{V}^k_{\mathcal{X}}(x))$. The larger the volume is required to hold the kNN, the lower the density. Hence, based on kNN, the radii of Dirichlet domains[1] in $\mathcal{X}$ centered at $\mathcal{Y}$ adapt to the local density. In that respect, density-based sampling corresponds to nearest neighbor queries with fixed k (i.e. kNN queries).

The direct implication of the above properties is that, if an indexing technique uses the above-defined $\mathcal{Y}$ as representative (pivot) set, then the inverted lists $\mathcal{L}_j$ associated with pivots x_j and defined by[2]

$$\mathcal{L}_j = \{x_i \in \mathcal{X} \mid d(x_i, x_j) \leq d(x_i, x_k) \quad \forall x_k \in \mathcal{Y}\}$$

are of constant size ($\mathbb{E}\,|\mathcal{L}_j| \simeq N/n$). Such a strategy is therefore profitable for indexing where obtaining short inverted lists is desirable for performance and a uniform partition of $\mathcal{X}$ into inverted lists guarantees this minimum.

However, preserving the density of representative samples and therefore creating a non-uniform geometrical partition of the data space is adverse at time of (geometrically) locating the query with respect to the dataset. At the time of locating the query, the relevance of a pivot $x_j \in \mathcal{Y}$ is related to its covering radius (e.g. $\mathrm{vol}(\mathcal{B}(x_j, \rho))$).

Further, given a fixed representation budget of pivots, the highest value for the lower bound for the distance from any query to any pivot is given by a geometrically uniform partition of the space. Emphasizing geometry (rather than density) therefore supports a more robust exclusion mechanism. For the same reason, it is also known that permutation-based indexing schemes that locate data by pivot activation benefit from a uniform partition of the data space by pivots [1].

[1] A Dirichlet domain is the generalization of a Voronoi region for high-dimensional spaces. Here, we look at subsets of data from $\mathcal{X}$ closer to a given point in $\mathcal{Y}$ than to any other point in $\mathcal{Y}$.

[2] Here, we allow $x_j \in \mathcal{L}_j$ since generically $\mathcal{Y} \subseteq \mathcal{X}$.

2.2 Geometry-Based Sampling

We therefore investigate the construction of a set of representatives $\mathcal{Y}$ based on geometric constraints. Dataset $\mathcal{X}$ is typically embedded into a domain $\Omega \subset \mathbb{R}^D$ that can be sampled using a D-dimensional regular lattice. Should any element from $\mathcal{X}$ fall into a simplex from the lattice, the center of that simplex (or the closest data from $\mathcal{X}$) may be taken as a representative. Basic examples of such a sampling include regular quantization of the coordinates of the original domain, or after applying some analysis such as PCA to discover (and potentially decimate) uncorrelated coordinates.

Representation Properties. Such a sampling strategy offers the advantage that the representative set $\mathcal{Y}$ lies close to a regular lattice and this regular structure may be exploited by the indexing.

To ensure geometric representation properties for $\mathcal{X}$, the criterion can be expressed as "$\mathcal{Y}$ covers uniformly the convex hull of $\mathcal{X}$", where the covering can be quantified by the k-center criterion:

$$\mathcal{Y} = \underset{\substack{\mathcal{S}\subset\mathcal{X} \\ |\mathcal{S}|=k}}{\operatorname{argmin}} \max_{x\in\mathcal{X}} d(x,\mathcal{S})$$

where, $d(x,\mathcal{S}) = \min_{x'\in\mathcal{S}} d(x,x')$. It is ensuring that data in $\mathcal{X}$ is never far from a sample in $\mathcal{Y}$. This is equivalent to minimizing the diameter of the Dirichlet domains built from $\mathcal{Y}$ of size k in $\mathcal{X}$. In that respect, geometric sampling corresponds to nearest neighbor queries with fixed range ε (i.e. range queries to uncover the εNN). In that case, pivots are associated to a fixed covering radius and inverted lists have lengths adapting to the local density.

3 Homogeneous Space Partitioning

3.1 Half Space Partitioning

In [7,9], we demonstrated that the local degree of the neighborhood graph built using the Half Space Partitioning (HSP) strategy [4] is an accurate proxy for the measurement of local intrinsic dimensionality. This is an important property for designing a geometrically efficient sampling strategy.

Algorithm 1 recalls the construction of the HSP, illustrated for the 2D case in Fig. 1. The HSP strategy partitions the hypersphere around every x_i into cones (see green dashed lines). In the HSP graph, each data point is connected (red edge) with its HSP neighbors and their mutual arrangement and the relationship with the Kissing number correlates their degree with the local dimensionality of the data [9]. Note that there is no upper bound for the distance value from x_i to the next selected HSP neighbor.

The construction of the HSP graph is highly parallel since the neighborhood of every point is computed independently of the rest. While this is a clear computational benefit and makes the HSP graph reproducible however the dataset

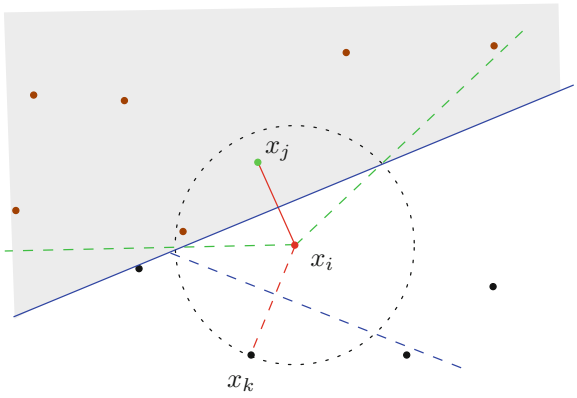

Fig. 1. HSP construction and discarding strategy. The (red) center data x_i chooses its closest (green) neighbor x_j as HSP neighbor and discards all data closer to x_j than to itself (shaded half-space). x_k will be selected as next closest neighbor (as symbolized by the dashed circle) and the next half-space (below the blue dashed line) discarded, until no neighbor of x_i remains (Color figure online)

Algorithm 1. HSP graph construction

```
1: procedure HSP(𝒳)                                   ▷ Half-space partitioning
2:     for every point x_i ∈ 𝒳 do
3:         while not all data in 𝒳 is discarded do
4:             Select the next nearest neighbor x_j ∈ 𝒳 not already discarded
5:             Add x_j as HSP-neighbor of x_i
6:             Discard any data x_k from 𝒳 that is closer to x_j than to x_i
```

is given, it makes the structure of the HSP graph unpredictable, apart from its properties arising from sphere packing.

In particular, no control is applied over the indegree of every node (the number of edges pointing *to* every node). As a result, there is no guarantee for a strong overlap of the HSP neighborhoods of 2 close points. Further, the specific structure of the HSP graph is sensitive to any data perturbation that would flip the order in which data appears as nearest neighbors of each other. In a setting where we use a point neighborhood as its representative, we would rather like to introduce correlation between neighborhoods of close points so as to:

- ensure that 2 close points share representatives (geometric consistency)
- obtain a compact, stable and sound representative sample of the data (statistical consistency)
- minimize the overall number of representatives

Here, we propose the "Hubness-HSP" (HubHSP for short) as a graph spanner over $\mathcal{X}$ supporting the selection of a representative set $\mathcal{Y}$. We first propose the rationale for its construction and then derive the actual construction algorithm.

We finally study and experimentally investigate the properties of the resulting HubHSP spanner for dataset sampling.

We wish to define the HubHSP as a structure that supports the selection of a representative set, while maintaining the favorable geometric properties of the HSP: x_j being selected as a neighbor of x_i means that x_j represents the vicinity of x_i and we wish to concentrate this representation into a given budget of representatives $\mathcal{Y}$. The base adaptation is therefore to install a control over the indegree of the nodes in the HubHSP. By enforcing nodes with high indegree, we create "centrality hubs[3]" that can be used to define representatives $\mathcal{Y}$ from the full set $\mathcal{X}$.

We therefore define a "hubness factor" h_j at every node x_j, which corresponds to its indegree during construction. Hence $\sum_j h_j = N$ and the challenge is to allocate h_j values so as to obtain concentrated hubs.

We build the graph following the aggregative compounding principle (see Fig. 2): a new data is matched with its HubHSP neighbors (line 9 in Algorithm 2) according to the HSP geometry while maintaining the most concentrated hubness by privileging existing hubs. Hence, at an intermediate stage, a data x_i is connected to the strongest current hub x_j from within its vicinity, and activates the HSP half-plane point discarding strategy.

Algorithm 2. Hubness HSP graph construction

```
1:  procedure HubHSP(𝒳)
2:      h_i ← 0  ∀i                                   ▷ Initialize hubness to 0
3:      Q.push(x_start)                               ▷ Initialize Q with x_start
4:      while Q is not empty do
5:          x_i ← Q.pop()                             ▷ Next data point in the chain
6:          Q.push(𝒱(x_i))                            ▷ Next data to consider in the chain
7:          C_i is the circle centered at x_i through its closest neighbor
8:          while not all data in 𝒳 is discarded do
9:              Select the neighbor x_j of x_i with maximum current hubness
10:             Add x_j as HSP-neighbor of x_i
11:             h_j ← h_j + 1                         ▷ Increase hubness of x_j
12:             x̃_j ← Proj_{C_i}(x_j)                 ▷ Project x_j onto C_i
13:             Discard any data x_l from 𝒳 that is closer to x̃_j than to x_i
```

We comment the main lines of Algorithm 2:

- Line 9: the current data x_i inspects a given vicinity $\mathcal{V}(x_i)$ (e.g. its 100-NN neighborhood) and finds the data x_j of current maximal hubness $h_j = \max_{x_k \in \mathcal{V}(x_i)} h_k$.
- Lines 10–11: x_j is added as neighbor to x_i by creating an edge (x_i, x_j) and therefore increasing the hubness (indegree) h_j of x_j.

[3] Here, centrality relates mainly to notion of degree centrality.

– Line 12: The natural distance-based selection in the HSP guarantees geometrical consistency [4]. This is not used anymore and to restore consistency, selected neighbors are projected onto the sphere C_i centered at x_i and containing the closest neighbor of x_i (blue circle in Fig. 2). In practice, this is done by proper normalization of vector $[x_i, x_j]$ into vector $[x_i, \tilde{x}_j]$ (see Annex).

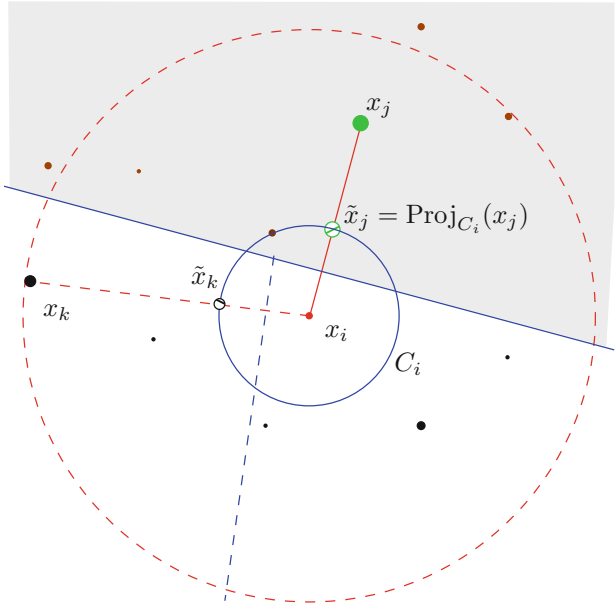

Fig. 2. HubHSP construction and discarding strategy. The current (red) center data x_i chooses its (green) neighbor x_j of highest hubness (size of the data point) as HubHSP neighbor from its vicinity $\mathcal{V}(x_i)$ (red dashed circle). It projects this data onto $\tilde{x}_j$ on the largest empty circle (blue circle) and discards all data closest to $\tilde{x}_j$ than to itself (shaded half-space). x_k will then be selected as next non-discarded neighbor of highest hubness and the next half-space (left to the blue dashed line, bisector of $[x_i, \tilde{x}_k]$) discarded, until no neighbor of x_i remains non-discarded (Color figure online)

The main practical adaptations from the HSP construction strategy are:

1. data is selected by *decreasing hubness* rather than *increasing distance*
2. because of 1. above, the selection of neighbors for x_i (line 4 in Algorithm 1) has to happen within the pre-defined vicinity $\mathcal{V}(x_i)$
3. because of 1. above, to maintain geometric consistency, points are projected onto a sphere of minimal radius around x_i before selection
4. since we now create a chain during the construction of the HubHSP (using Q), a starting point has to be defined.

The first and main benefit of this adaptation is the creation of a hubness index h_j per datum (node in the HubHSP graph). The hubness index h_j is the indegree of node x_j in the HubHSP graph. h_j counts how many data x_i have x_j as HubHSP neighbor. A node with high hubness is therefore an interesting candidate for the representative subset. This provides a sound and natural strategy for the selection of $\mathcal{Y}$ by simply selecting nodes in decreasing order of their indegree.

As a result, the HubHSP graph combines two properties. From its inheritance from the HSP process, the outdegree of every node reflects the local geometry (intrinsic dimensionality) of the data [9]. Through the hubness, the indegree of each node is now correlated with the statistical properties of the data.

Since in practice we need to define (limit) the vicinity $\mathcal{V}(x_i)$ from where the HubHSP neighbors are selected (line 9 in Algorithm 2), the construction of this set impacts the resulting properties of the HubHSP graph.

- if $\mathcal{V}(x_i) = \mathcal{V}^k_{\mathcal{X}}(x_i)$, the kNN neighborhood of x_i in $\mathcal{X}$, the span of this set is driven by the local density, as discussed above. Hence, the kNN-based HubHSP graph reflects the local density of data via arc lengths, on top of reflecting its geometry via outdegree.
- if $\mathcal{V}(x_i) = \mathcal{V}^\varepsilon_{\mathcal{X}}(x_i)$, the εNN neighborhood of x_i in $\mathcal{X}$, the span of this set is immune from the local density and it is the indegree of every neighbor that reflects this density.

Hence, the HubHSP graph adds to the HSP graph the encoding of the local density either via arc lengths (kNN) or indegree (εNN).

3.2 Complexity

The base complexity of the HubHSP construction algorithm is $O(N^2D)$. It mimics that of the computation of any neighborhood graph as it is dominated by selection of candidate neighbors (line 9 in Algorithm 2). Such a complexity may classically be reduced by a pre-indexing of these neighborhoods. In Sect. 4, we present results against baselines whose base complexities are of the same order.

3.3 Generic Metric Spaces

Our discussion and illustration have been concerned with metric space (Ω, d) where $\Omega \subset \mathbb{R}^D$ and $d(.,.)$ is the Euclidean distance function. All definitions provided here rely on the existence of a proper distance function and therefore do generalize to other metric spaces. The precise study of the properties obtained when constructing the HubHSP in these metric spaces is out of the scope of this paper and is left for an extension.

4 Experiments

We now experiment under various conditions and compare to relevant baselines.

4.1 Dataset

To highlight the properties of our proposal, we use data with various properties in terms of density and dimension D. As a base reference, we generate 2 artificial dataset with uniform distribution $\mathcal{U}^{100K\times 2}$ and $\mathcal{U}^{100K\times 10}$, containing 100'000 data of dimension $D = 2$ and $D = 10$, respectively. Note that in this case, the dataset of dimension 10 with 100'000 data is rather sparse.

To depart from the uniform distribution, we generate 2 dataset $\mathcal{N}^{100K\times 2}$ and $\mathcal{N}^{100K\times 10}$ with the same parameters but from a centered normal distribution. While uniformity makes the density of the data the same at every point in space, the Normal distribution induces an exponential variation of the density across the space.

As a more realistic dataset, we use the 500'000 first data of the ANN SIFT (base set) benchmark [8]. In this case $D = 128$, inducing a very sparse set. We also use a dataset of Flow Cytometry data containing $N = 470'995$ $D = 18$-dimensional data. This data is known by definition to aggregate in dense localized clusters (see Fig. 3 for a 2D glance). Its distribution is therefore far from uniform with large unpopulated parts of the space.

In all cases, we set the size n of the sample to 1% of the original size N. We fixed $k = 1000$ and $\varepsilon = 20$ to create the base neighborhoods ($\mathcal{V}^k_{\mathcal{X}}(x_i)$ and $\mathcal{V}^{\varepsilon}_{\mathcal{X}}(x_i)$ respectively) over which the HubHSP graph is built.

4.2 Baselines

Random. As discussed above, a uniform sampling of the data indices ensures the preservation of the statistical properties (density) of the data into the sample.

Farthest First Traversal (FFT). In contrast, this geometrical strategy aims at spreading the representative set across the dataset by approximating the k-center problem [6]. Using this strategy it is expected that the representative samples lie close to a regular grid.

Note that due to the concentration of distance phenomenon, this strategy loses its rationale in high dimensions.

k-means ++ [2] adds a random component to the above FFT strategy by making it most likely but not a strict choice, depending on the density of the data. k-means ++ is therefore interesting since it offers theoretical bounds in representation and mixes geometrical and statistical constraints, as we aim to do here.

4.3 Measures and Results

We use the following measures to assess the characteristics of our proposed sampling. Results are reported in Table 1.

Table 1. Evaluation measures across dataset and techniques. Top section: empty sphere. Middle section: lengths of inverted lists (standard deviation). Bottom section: Maximum distance. Values between parenthesis are standard deviation values

	$\mathcal{U}^{100K\times10}$	$\mathcal{U}^{100K\times2}$	$\mathcal{N}^{100K\times10}$	$\mathcal{N}^{100K\times2}$	SIFT$^{500K\times128}$	FlowCyto$^{471K\times18}$
Random	100.95 (16.72)	**3.19 (1.69)**	182.99 (39.85)	**7.47 (7.24)**	250.35 (47.76)	146.48 (47.64)
FFT	147.51 (4.85)	5.75 (0.54)	330.13 (12.77)	19.37 (5.03)	359.20 (6.33)	321.67 (26.19)
k-means++	107.97 (14.32)	4.12 (1.36)	200.93 (39.88)	13.20 (10.88)	265.47 (38.37)	178.04 (64.88)
HHSP (kNN)	108.76 (10.51)	4.25 (1.78)	170.28 (18.32)	10.69 (10.42)	244.77 (35.84)	146.37 (38.61)
HHSP (eNN)	**94.66 (15.88)**	3.32 (1.45)	**151.47 (31.20)**	9.91 (4.20)	**203.26 (58.11)**	**115.20 (40.76)**
Random	46.47	53.76	59.60	54.27	86.53	66.34
FFT	76.04	**23.05**	340.66	149.93	329.60	530.15
k-means++	47.97	36.15	77.93	69.09	112.12	90.99
HHSP (kNN)	33.25	35.34	**24.21**	**37.97**	60.11	53.05
HHSP (eNN)	**33.17**	54.26	27.20	51.65	**56.91**	**36.88**
Random	100.38 (18.96)	3.17 (1.71)	180.76 (43.61)	7.55 (8.15)	247.86 (53.62)	145.92 (51.40)
FFT	106.68 (17.56)	**2.51 (1.00)**	240.60 (37.13)	7.79 (3.14)	296.86 (40.41)	218.95 (34.87)
k-means++	100.6 (18.52)	2.78 (1.34)	183.64 (41.57)	6.41 (3.88)	249.60 (50.90)	146.72 (44.19)
HHSP (kNN)	98.51 (19.36)	2.69 (1.16)	**175.17 (47.62)**	**6.32 (5.59)**	245.89 (52.23)	147.88 (59.17)
HHSP (eNN)	**96.85 (19.57)**	3.17 (1.62)	175.90 (48.16)	7.30 (12.11)	**239.22 (56.73)**	**138.20 (53.47)**

The empty sphere measure (top section) quantifies the uniformity of the sampling by measuring the diameter of the largest empty sphere lying between samples. In practice it is the maximum distance between 2 neighboring samples.

Since we wish an equipartition of the space by samples, the smaller this value is, the better the quality of the sample. We report the mean and also measure uniformity of this allocation by reporting the standard deviation (between parenthesis).

We see that in the most basic conditions ($\mathcal{U}^{100K\times 2}$) all sampling strategies perform similarly. When the dimension increases (e.g. $\mathcal{U}^{100K\times 10}$), the data becomes sparser and geometrical techniques (such as FFT) fail. Our proposal is able to consistently reduce the value of the measure while keeping the variance at a comparable level.

The length of inverted lists (middle section) is an indicator of the uniformity of the allocation of representative to the data. In practice, since we use Dirichlet domains to define the lists, the average list length is simply the ratio between the size of the data and the sample ($\mathbb{E}\,|\mathcal{L}_j| = N/n$) so only the standard deviation is reported. The smaller this value, the more uniform the partition is.

We clearly see the same trend of lower variance in the length of inverted lists and therefore more stability in the allocation of representative data.

The maximum distance (bottom section) between a data and its representative is rather based on the data. It is a geometric indicator of how well every data is represented by the sample. Ideally, every data should find a representative in its vicinity so again, the smaller this value is, the better. We report the mean and also measure uniformity of this allocation by reporting the standard deviation (between parenthesis).

This measure shows that the HubHSP hubness allocates representatives closer to each data than other strategies. This is understood by the ability of the HubHSP to exploit better the statistical and geometrical properties of the data to allocate better a fixed budget of n representative data. This is made clear in the most adverse setting of high-dimensional non-uniform data (which corresponds to real dataset).

Figure 3 proposes a visual intuition of the allocation of representatives in low-dimensional non-uniform data. The resulting samples (red points) are shown over the data (green points) for all baselines and for the HubHSP. An ideal sampling should show regularity (to avoid redundancy) and respect the data density.

Whereas random sampling (top left) is inefficient by allocating redundant representative samples, the FFT (top right) is inefficient by being blind to the local density. k-means ++ (lower left) proposes an adequate mix of statistical and geometrical sampling but clearly the HubHSP (lower right) adds a form of regularity that removes local density artifacts due to random sampling and explains the effectiveness in terms of geometrical partitioning (Dirichlet domains) of the data.

Finally, Fig. 4 shows an histogram of the corresponding hubness values h_j. A very large majority of these values are zero, which demonstrates the ability

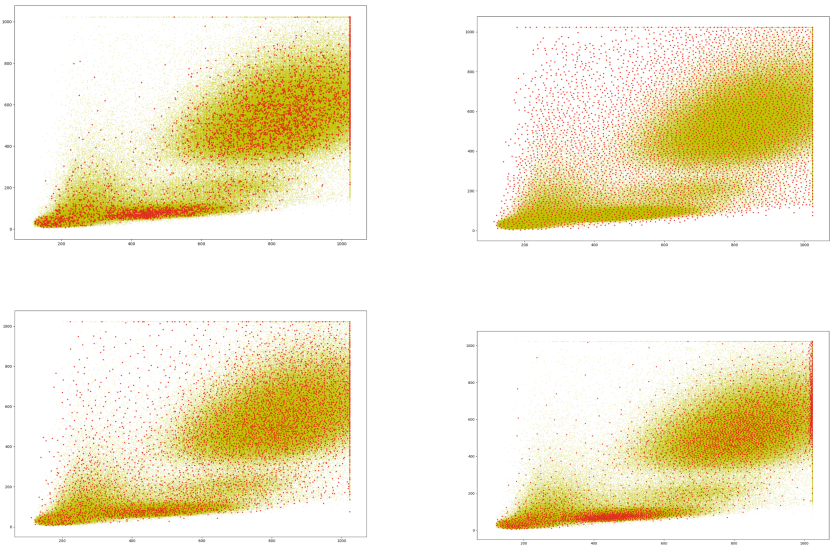

Fig. 3. Sampling strategies by the baselines and the HubHSP over a 2D slice of the FlowCyto dataset (FlowCyto$^{471k\times 2}$) as a low-dimensional non-uniform example. In each scatter plot, the dataset is shown in green and selected representatives are shown in red. [top left] Random uniform, [Top right] FFT, [Lower left] k-means ++, [Lower right] HubHSP (ours) (Color figure online)

of the HubHSP to concentrate its indegree into only a minority of large values (since $\sum_j h_j = N$). This indicates that only a small percentage of data in $\mathcal{X}$ then compete for entering $\mathcal{Y}$.

5 Conclusion

Subsampling a finite dataset may be considered from either a statistical or geometrical perspectives. Classical strategies focus on either of these. Based on the capability of the HSP graph to correlate with the local intrinsic dimensionality we proposed the HubHSP to generate a sound data selection criterion combining geometrical and statistical properties.

We demonstrate the ability of the HubHSP graph construction algorithm as a modification of the HSP graph construction to indicate a sound and stable selection of data as representative. We compare with classical selection algorithm and show that the HubHSP is able to create a more robust and effective sampling by a better exploitation of geometrical constraints on top of statistical sampling.

More generally, this work relates the ability of graph spanners to mirror and combine geometrical and statistical properties of non-uniform point clouds in high dimensions. In [9] diffusion over neighborhood graphs was used to exhibit that structure exploiting the link between connectivity (resp degree) and cen-

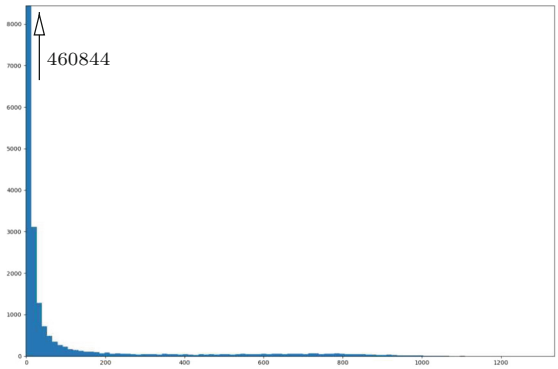

Fig. 4. Hubness for the 2D slice of the FlowCyto dataset (FlowCyto$^{471k\times 2}$) shown in Fig. 2 [Lower right]. Only about 2.2% of the values are non-zero.

trality. There is much to explore in this interplay of data analysis methods and data modeling techniques to particularize subsets of dataset.

Acknowledgments. This work is partly funded by the Swiss National Science Foundation under grant number 207509 "Structural Intrinsic Dimensionality".

Annexes

HubHSP Projection. The HSP selects its neighbors based on increasing distance after discarding half-planes. Since the neighbors selected by the HubHSP can occur in random order of their distance values from the central point x_i, it is critical to consider them as projected over a common sphere centered at x_i.

The most canonical choice is the sphere C_i including the first neighbor x_l of x_i. Note $\rho_i = d(x_l, x_i)$ its radius (the distance between x_i and its closest neighbor), then a point x_j is projected as $\tilde{x}_j$ onto C_i by:

$$\tilde{x}_j = \mathrm{Proj}_{C_i}(x_j) = \operatorname*{argmin}_{x \in C_i} d(x, x_j) = x_i + \rho_i \frac{x_j - x_i}{d(x_j, x_i)}$$

Main Mathematical Symbols

Ω	Ambient space
$\mathcal{X}$, $\mathcal{Y}$	Main dataset, representative set
$[\![N]\!]$	Set of indices $\{1 \cdots N\}$
$d(.,.)$	distance function
$\mathcal{B}(x, \rho)$	Ball centered at x of radius ρ
$\mathrm{Proj}_C(x)$	Projection of x onto C
$f_{\mathcal{X}}$	True pdf of the dataset
$\hat{f}_{\mathcal{X}}$	Empirical density of the dataset
$\mathcal{V}_{\mathcal{X}}^k(x)$	k-closest neighbors of x in $\mathcal{X}$
$\mathcal{V}_{\mathcal{X}}^{\varepsilon}(x)$	ε-neighbors of x $(= \mathcal{B}(x, \varepsilon) \cap \mathcal{X})$
$\mathcal{L}_j$	Inverted list for x_j
$\mathbf{E}X$	Expectation of variable X

References

1. Amato, G., Esuli, A., Falchi, F.: A comparison of pivot selection techniques for permutation-based indexing. Inf. Syst. **52**, 176–188 (2015). https://doi.org/10.1016/j.is.2015.01.010
2. Arthur, D., Vassilvitskii, S.: K-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, pp. 1027–1035. Society for Industrial and Applied Mathematics, USA (2007)
3. Bustos, B., Navarro, G., Chávez, E.: Pivot selection techniques for proximity searching in metric spaces. Pattern Recogn. Lett. **24**, 2357–2366 (2003)
4. Chavez, E., et al.: Half-space proximal: a new local test for extracting a bounded dilation spanner of a unit disk graph. In: Anderson, J.H., Prencipe, G., Wattenhofer, R. (eds.) OPODIS 2005. LNCS, vol. 3974, pp. 235–245. Springer, Heidelberg (2006). https://doi.org/10.1007/11795490_19
5. Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.L.: Searching in metric spaces. ACM Comput. Surv. **33**(3), 273–321 (2001)
6. Dasgupta, S., Long, P.M.: Performance guarantees for hierarchical clustering. J. Comput. Syst. Sci. **70**, 555–569 (2005). Farthest First Traversal for Pivot Selection
7. Hoyos, A., Ruiz, U., Marchand-Maillet, S., Chávez, E.: Indexability-based dataset partitioning. In: Amato, G., Gennaro, C., Oria, V., Radovanović, M. (eds.) SISAP 2019. LNCS, vol. 11807, pp. 143–150. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32047-8_13
8. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Trans. Pattern Anal. Mach. Intell. **33**(1), 117–128 (2011)
9. Marchand-Maillet, S., Pedreira, O., Chávez, E.: Structural intrinsic dimensionality. In: Reyes, N., et al. (eds.) SISAP 2021. LNCS, vol. 13058, pp. 173–185. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-89657-7_14
10. Ruiz, G., Chávez, E., Ruiz, U., Tellez, E.S.: Extreme pivots: a pivot selection strategy for faster metric search. Knowl. Inf. Syst. **62**(6), 2349–2382 (2020). https://doi.org/10.1007/s10115-019-01423-5
11. Terrell, G.R., Scott, D.W.: Variable kernel density estimation. Ann. Stat. **20**(3), 1236–1265 (1992)