Taming the curse of dimensionality for perturbed token identification

Olga Assainova, Jérémy Rouot, Ehsan Sedgh-Gooya

Taming the curse of dimensionality for perturbed token identification

Olga Assainova
L@ISEN, Isen Brest
Brest, France
olga.assainova@yncrea.fr

Jérémy Rouot
L@ISEN, Isen Brest
Brest, France
jeremy.rouot@yncrea.fr

Ehsan Sedgh-Gooya
L@ISEN, Isen Brest
Brest, France
ehsan.sedgh-gooya@yncrea.fr

***Abstract*—In the context of data tokenization, we model a token as a vector of a finite dimensional metric space E and given a finite subset of E, called the token set, we address the problem of deciding whether a given token is in a small neighborhood of an other token. We derive conditions to characterize the nearest token of a given one and show that these conditions are fulfilled asymptotically as the dimension of E tends to infinity. Whereas the classical nearest neighbor search is inefficient to solve such problem, we propose a new probabilistic algorithm, which becomes efficient if the dimension of E is large enough.**

***Index Terms*—Tokenization, Big data, Algorithmic probability, Nearest neighbor search**

INTRODUCTION

With the explosion of sensitive data, many standards emerge to secure information and reduce the number of incidents that may occur during an inappropriate access to a database [6]. Tokenization consists in associating to a sensitive data an identifier (called token) that has non-external or exploitable meaning related to the data that it corresponds. While tokenization seems to be a reliable method for data obfuscation, identifying whether a given token belongs to a token set has an impact on the performance of the application.

We model the token space as a metric space (E, d) of finite dimension n and the token set $\mathcal{T}$ is a finite subset of E. We will consider the discrete case where $\mathcal{T} \subset [\![0, c]\!]^n$, $c \in \mathbb{N}$. The distance d can be induced by the Euclidean norm $\|\cdot\| = \sqrt{(\cdot, \cdot)}$. This particular case where E is a normed vector space, instead of general metric space, allows to consider orthogonal projections. A token $\tilde{\tau}$ is considered as a neighbor of a token $\tau \in \mathcal{T}$, and we note $\tilde{\tau} \sim \tau$, when $d(\tau, \tilde{\tau})$ is small enough. Given $\mathcal{T}$ and $\tilde{\tau}$, we would like to find, if it exists, a neighbor $\tau \in \mathcal{T}$ of $\tilde{\tau}$.

This falls into the problem of similarity query in a metric space. This problem can be solved by nearest neighbor search (NNS) algorithms and the main bottleneck remains the so-called curse of dimensionality [2], [3]. When the dimension n is large, the curse of dimensionality means, under reasonable assumptions on the token distribution, that the ratio between the distance of the nearest and the farthest neighbors is close to 1 [1]. We exploit this property to compute the nearest neighbor in $\mathcal{T}$ of a given token, which is considered to be a perturbation of a token in $\mathcal{T}$. We propose a different approach than NNS based on orthogonal projections filtering. The complexity of NNS algorithms is characterized by the number of distance computations and memory limitation. We will use none of these complexities but rather time complexity which is more adapted for our case, where we assume to have enough memory to stock and sort the token database. This operation is done only one time – at the beginning of the oracle – and the benefit over other algorithms appears when the oracle is called quite a number of times.

Section I introduces the model of the token database and defines a metric to characterize a perturbed token $\tilde{\tau}$ of a given token $\tau \in \mathcal{T}$. We present a naive NNS to compute such neighbor $\tau \in \mathcal{T}$ and that will be the benchmark of our new algorithm presented in Section II. We give mathematical conditions on the cardinal of $\mathcal{T}$ and the dimension of E to ensure that the probability that our conditions are satisfied is closed to 1. Section III is devoted to preliminary numerical results.

I. MATHEMATICAL FORMULATION AND CONCEPTS

a) Notations: Throughout the article, $(E, (\cdot \mid \cdot))$ is an inner product space over the field of real numbers of finite dimension n. The induced norm of a vector $x \in E$ is denoted by $\|x\| = \sqrt{(x \mid x)}$. E can also be seen as a metric space, induced by the distance defined by $d(x, y) = \|x - y\|$ for $x, y \in E$. The components of a vector $x \in E$ are written in superscript as $x^{(1)}, \dots x^{(n)}$. The ℓ-norm, $\ell \in \mathbb{N}$, of a vector $x \in E$ is $\|x\|_\ell = \left(\sum_{i=1}^n |x^{(i)}|^{1/\ell}\right)^\ell$ and the infinity norm is $\|x\|_\infty = \max_{i=1\dots n} |x^{(i)}|$. The distance associated to the ℓ-norm will be denoted by d_ℓ.

b) Model: The tokens $x_1, \dots, x_N$ are realizations of a discrete random vector X valued in $[\![0, c]\!]^n \subset E$. This is equivalent to say that each x_i, $i = 1 \dots N$ is a realization of a random variable X_i, $i = 1 \dots N$, X_i's being independent and identically distributed (i.i.d.) with the same law as each component of X.

We decompose a vector $x \in E$ into p parts as follows. Choose $n_1, \dots, n_p \in \mathbb{N}$ such that $n = n_1 + \dots + n_p$ and define for $j = 1 \dots p$, the projections $\pi_j : E \to \mathbb{R}^{n_j}$ by $\pi_j(x_i) = (x_i^{(s_j+1)}, \dots, x_i^{(s_{j+1})})$ and $s_j = \sum_{k=1}^{j-1} n_k$ (with the

Calculations were performed using HPC resources from DNUM CCUB (Centre de Calcul de l'Université de Bourgogne).

convention $s_1 = 0$). Hence, the components of a vector $x \in E$ in the canonical basis are the components of the concatenation of the vectors $\pi_j(x)$, $j = 1 \dots p$.

c) Necessary conditions: Assume that for all $n > 0$ and $1 \leq p \leq n$, there exists $\varepsilon > 0$ such that

(A1) $\|x_1 - x_0\| < \varepsilon$,

(A2) for all $k \in \{2, \dots, N\}$, there exists $j \in \{1, \dots, p\}$ such that $|\|\pi_j(x_k)\| - \|\pi_j(x_0)\|| \geq \varepsilon$.

A quick look at these conditions gives that the nearest neighbor of x_0 in the token space $\{x_1 \dots x_N\}$ is x_1. Indeed, we have $d(x_0, x_1) < \varepsilon$ and for $k \in \{2, \dots, N\}$ and $j \in \{1, \dots, p\}$, $d(x_0, x_k) \geq d(\pi_j(x_0), \pi_j(x_k)) \geq \varepsilon$.

Our new algorithm is based on the following theorem.

Theorem 1. *If* $(A1), (A2)$ *hold, then we have*

$$\underset{k \in \{1,\dots,N\}}{\operatorname{argmin}} \max_{j_k \in \{1,\dots,p\}} |\|\pi_{j_k}(x_k)\| - \|\pi_{j_k}(x_0)\|| = 1. \quad (1)$$

Proof. Let $e_1, \dots, e_n$ be an orthogonal basis of $(E, (\cdot \mid \cdot))$. For $x \in E$, the vector $\pi_j(x)$ denotes the orthogonal projection of a vector x on $E_j = \operatorname{span}(e_{s_j+1}, \dots, e_{s_{j+1}})$. From $(A1)$, we get $\|\pi_j(x_0 - x_1)\| < \varepsilon, \forall j = 1 \dots p$. For each $k \in \{1, \dots, N\}$, we compute $j_k \in \{1, \dots, p\}$ such that $|\|\pi_{j_k}(x_k)\| - \|\pi_{j_k}(x_0)\||$ is maximal. Once we have such projection π_{j_k}, suppose that $m \in \{2 \dots N\}$ satisfies

$$|\|\pi_{j_m}(x_m)\| - \|\pi_{j_m}(x_0)\|| = \min_{1 \leq k \leq N} |\|\pi_{j_k}(x_k)\| - \|\pi_{j_k}(x_0)\||.$$

Then, we have

$$\begin{aligned} |\|\pi_{j_m}(x_m)\| - \|\pi_{j_m}(x_0)\|| &\leq |\|\pi_{j_1}(x_1)\| - \|\pi_{j_1}(x_0)\|| \\ &\leq \|\pi_{j_1}(x_1 - x_0)\| \\ &< \varepsilon, \end{aligned}$$

and, for all $\ell \in \{1 \dots p\}$,

$$|\|\pi_\ell(x_m)\| - \|\pi_\ell(x_0)\|| \leq |\|\pi_{j_m}(x_m)\| - \|\pi_{j_m}(x_0)\||.$$

Hence, we deduce

$$|\|\pi_\ell(x_m)\| - \|\pi_\ell(x_0)\|| < \varepsilon, \quad \forall \ell \in \{1 \dots p\},$$

which contradicts the assumption $(A2)$. Therefore the minimum in (1) is attained at $k = 1$ and this concludes the proof. □

II. A new approach for finding a perturbed token in a token set

A. A variant of the nearest neighbor search

We cannot implement directly the optimization problem given in Theorem 1 because it would require for a given token x_0 to compute the quantities $|\|\pi_j(x_k)\| - \|\pi_j(x_0)\||$ for all $k = 1 \dots N$ and $j = 1 \dots p$ and it offers no benefit over a naive NNS.

We propose in Algorithm 1 a new method to decide whether a given token x_0 is a neighbor of a token of $\mathcal{T}$. The main point is not to break the dimensionality but to construct a reduced set of tokens satyfying the conditions $(A1) - (A2)$ using the functions $x \to \|\pi_j(x)\|$, $j = 1 \dots p$.

```
Input:
  • The dimension $n$ of $E$,
  • a token set $\mathcal{T} = \{x_1, \dots, x_N\}$ and a token $x_0 \in E$
  • the number $p$ of projections $\pi_j$,
  • the sorted $N$-tuples $\mathcal{P}_j$, $j = 1 \dots p$ composed of the
    numbers $\|\pi_j(x_k)\|$, $k = 1 \dots N$,
  • the permutations $\sigma_j$, $j = 1 \dots p$ obtained from the
    sort of $\mathcal{P}_j$ ($\sigma_j(k)$ is the position of $\|\pi_j(x_k)\|$ in $\mathcal{P}_j$),
  • an integer $\eta$
Output: Return the nearest token of $x_0$ in $\mathcal{T}$ provided
        $\eta$ is well chosen.
for $j = 1 \dots p$ do
    insert $\|\pi_j(x_0)\|$ at the right place in $\mathcal{P}_j$;
    update $(\sigma_j(k))_{k=1 \dots N}$ ;
end
$I = \cap_{j=1}^{p} \{ \sigma_j^{-1}(\max(\sigma_j(0) - \eta, 0)), \dots,$
                $\sigma_j^{-1}(\min(\sigma_j(0) + \eta, N)) \} \setminus \{x_0\}$ ;
return The nearest token of $x_0$ taken in $I$;
```

Algorithm 1: Nearest token search with projective discrimination.

We have the following proposition.

Proposition 2. *Given any token set satisfying the assumptions* $(A1) - (A2)$ *and any integer* η, *Algorithm 1 is exact for* $p = 1$.

a) Discussion: The main drawback is that the algorithm is not exact if η is too small, except for the case $p = 1$, where the algorithm is exact for any η provided $(A1) - (A2)$ are satisfied. This is illustrated by the following situation where $N = 3, n = p = 2$ and $x_0 = (1, \varepsilon)$, $x_1 = (0, 0)$, $x_2 = (\alpha^2, \varepsilon/2)$, $x_3 = (2, \beta^2)$, and $\alpha, \beta \in \mathbb{R}$ and $\varepsilon > 0$. If α, β are large enough, we would like the Algorithm 1 to return x_1. We have $\mathcal{P}_1 = (0, 1, 2, \alpha^2)$, $\sigma_1 = (1, 0, 3, 2)$, $\mathcal{P}_2 = (0, \varepsilon/2, \varepsilon, \beta^2)$, $\sigma_2 = (2, 0, 1, 3)$ and for $\eta = 1$ we have $I = \{3\}$, which yields the wrong index.

A naive NNS computes all the distances $d(x_k, x_0)$, $k = 1 \dots N$ and keep the smallest one. Note that even if we use an adapted data structure to allocate $\mathcal{T}$ (dictionary, Adelson-Velsky and Landis tree, self-organising binary search tree ...), we would still need to compute all the distances for a new token x_0. Algorithm 1 begins with sorting the lists $(\|\pi_j(x_k)\|)_k$ to construct $\mathcal{P}_j$ for $j = 1 \dots p$ only once (assuming the token set $\mathcal{T}$ remains unchanged for further calls). This step is crucial and this memory allocation is an other drawback of our algorithm.

The idea of Algorithm 1 is to perform a naive NNS in a set I, which is drastically smaller then the initial token set $\mathcal{T}$, provided the parameter η is small. The crucial and difficult point is to have an estimation of the parameter η, and Section II tends to start this analysis using probability theory. Time comparison between Algorithm 1 and a naive NNS are given

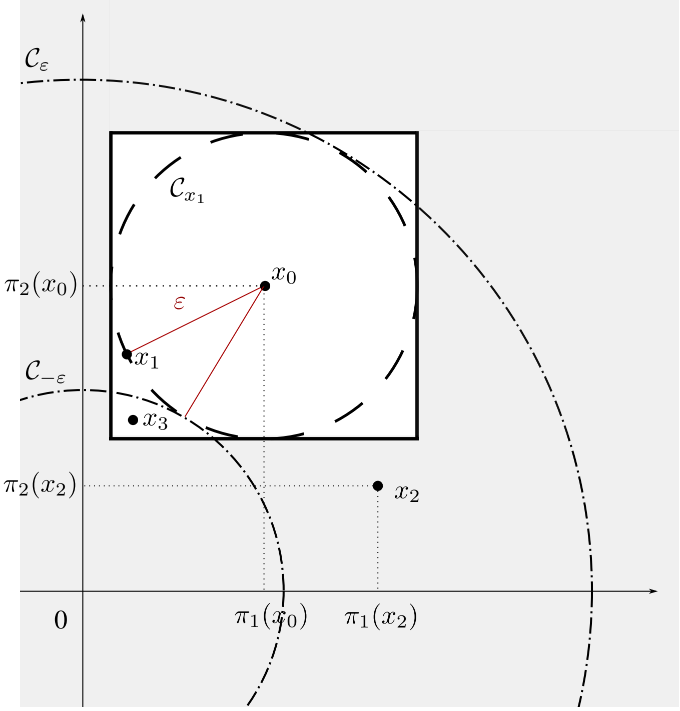

Fig. 1. Illustrative example for the weakness of the assumption $(A2)$ if $p > 1$, compared with $p = 1$, required by Theorem 1. The assumption $(A2)$ is satisfied for $p = 2$, but not for $p = 1$ due to the token x_2.

in Section III.

The number of projections p is an interesting parameter. If $p = 1$, we do not recover the naive NNS but the algorithm becomes *exact* if the assumptions (A1)-(A2) are sastified. The assumptions $(A1) - (A2)$ for $p > 1$ are weaker, in terms of probability, than for $p = 1$. This is illustrated by Fig. 1 for the case $n = 2$: the nearest neighbor of x_0 is x_1. If $p = 1$, the assumption $(A2)$ is satisfied if the tokens $x_k,\ k = 2 \dots N$ are outside the annulus delimited by the circles $\mathcal{C}_{-\varepsilon}$ and $\mathcal{C}_\varepsilon$. If $p > 1$, the assumption $(A2)$ is satisfied if the tokens $x_k,\ k = 2 \dots N$ are outside the square centered at x_0 of 2ε side – represented by the white region. For instance, the point x_2 satisfies the assumption $(A2)$ for $p = 2$ but not for $p = 1$, while $(A2)$ is satisfied by the point x_3 for $p = 1$ but not for $p = 2$. Computations shall be made to precise the relation between the assumption $(A1) - (A2)$ and the number p.

Another interesting perspective is to consider $\mathcal{P}_j = (f(x_k))_{k=1\dots N}$, $j = 1 \dots p$ where f is a given function (we formulated our algorithm for $f = \|\cdot\|$) to guarantee that the parameter η is small enough.

b) Complexity: The sort of the lists $\mathcal{P}_j$, $j = 1 \dots p$ can be done offline, and we do not take it into account in the complexity analysis. Contrary to NNS, the number of distance computations of our algorithm is not relevant, we will deal with time complexity.

If η^* corresponds to the optimal value of the parameter η for which the algorithm 1 returns the nearest token x_1, the time complexity of Algorithm 1 is in $O(p\eta^*)$ since the time complexity of the intersection between sets of cardinals $2\eta^*+1$ is $O(\eta^*)$. The crucial point is then to determine in the average case or worst case an estimate on η^* for $p > 1$.

B. Probability computation

We consider i.i.d. scalar-valued discrete random variables $X_i^{(k)}$, $k = 1 \dots n, i = 1 \dots N$. The distribution of X_i is defined from the joint probability mass function of $X_i^{(1)}, \dots, X_i^{(n)}$, that is:

$$P(X_i = x_i) = P(X_i^{(1)} = x_i^{(1)}, \dots, X_i^{(n)} = x_i^{(n)}). \quad (2)$$

Since the random variables $X_i^{(k)}$, $k = 1 \dots n$, $i = 1 \dots N$ are independent, we have

$$P(X_i^{(1)} = x_i^{(1)}, \dots, X_i^{(n)} = x_i^{(n)}) \\ = P(X_i^{(1)} = x_i^{(1)}) \dots P(X_i^{(n)} = x_i^{(n)}).$$

For the application, the number of tokens N is fixed and our aim is to prove that

$$P(\exists i \neq j,\ d(X_i, X_j) \leq \varepsilon)$$

is small, where d is a distance on E. We will treat the case where the distance d is the Euclidean distance, or induced by the L^1 and L^∞ norm.

a) Scalar case: The random variables $X_i,\ i = 1 \dots N$ are valued in $[\![0, c]\!]^n$. We recall properties to manipulate discrete random variables, see [4] for details.

Definition 3. *Let U be a discrete random variable valued in $[\![0, c]\!]$. The probability generating function of U is the polynomial function*

$$G_U(z) = \sum_{k=0}^{c} P(U = k)\, z^k.$$

Lemma 4. *The probability generating function of the sum $U + V$ of two independent discrete random variables U, V is $G_{U+V}(z) = G_U(z)\, G_V(z)$.*

Lemma 5. *Given two i.i.d. random variables U, V valued in $[\![0, c]\!]$, the probability generating function of the variable $|U - V|$ is $G_{|U-V|}(z) = \sum_{k=0}^{c} P(|U - V| = k)\, z^k$ with*

$$P(|U - V| = 0) = \sum_{i=0}^{c} p_i^2 \quad (3)$$

and

$$\begin{pmatrix} P(|U-V| = 1) \\ \vdots \\ P(|U-V| = c) \end{pmatrix} = 2\, H(p_1, \dots, p_c) \begin{pmatrix} p_0 \\ \vdots \\ p_{c-1} \end{pmatrix}, \quad (4)$$

where $p_k = P(U = k) = P(V = k)$, $k = 0 \dots c$ and $H(p_1, \dots, p_c)$ is the Hankel matrix associated to the probabilities $p_1, \dots, p_c$ defined by

$$H(p_1, \dots, p_c) = \begin{pmatrix} p_1 & p_2 & p_3 & \dots & p_c \\ p_2 & p_3 & \dots & p_c & 0 \\ \vdots & & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & & \vdots \\ p_c & 0 & \dots & \dots & 0 \end{pmatrix}.$$

b) Euclidean norm:

Proposition 6. *Let X_i, X_j be two random variables valued in $[\![0, c]\!]^n$. Then, for $m \in [\![0, \dots, \lfloor c\sqrt{n} \rfloor]\!]$, we have*

$$P(\|X_i - X_j\|_2 \le m) = \sum_{k=0}^{m^2} [z^k] \left(G_{(U-V)^2}(z) \right)^n, \tag{5}$$

where $[z^k]Q(z)$ denotes the coefficient of the monomial of degree k of the polynomial $Q(z)$, that is $[z^k]Q(z) = k! \dfrac{\mathrm{d}^k Q(z)}{\mathrm{d}z^k}_{|z=0}$, and U, V are random variables with the same distribution than the marginal distribution of X_i, X_j.

Proof. Since $(X_i^{(k)} - X_j^{(k)})^2$, $k = 1 \dots n$ are independent and have the same law as $(U - V)^2$, the distribution of U, V being the distribution of $X_i^{(k)}, X_j^{(k)}$ then by Lemma 4 we have for $m \in [\![0, \dots, \lfloor c\sqrt{n} \rfloor]\!]$,

$$\begin{aligned} & P(\|X_i - X_j\|_2 \le m) \\ & \quad = P\left(\sum_{k=0}^{n} \left(X_i^{(k)} - X_j^{(k)} \right)^2 \le m^2 \right) \\ & \quad = \sum_{k=0}^{m^2} [z^k] \left(G_{(X_i^{(1)} - X_j^{(1)})^2}(z) \dots G_{(X_i^{(n)} - X_j^{(n)})^2}(z) \right), \end{aligned}$$

which yields the result. □

To compute the polynomial $G_{(U-V)^2}(z)$ in Proposition 6, observe that,

$$\begin{aligned} G_{(U-V)^2}(z) &= \sum_{k=0}^{c^2} P((U - V)^2 = k)\, z^k \\ &= \sum_{k=0}^{c} P(|U - V| = k)\, z^{k^2}, \end{aligned} \tag{6}$$

and $P(|U - V| = k)$, $k = 0 \dots c$ can be found using Lemma 5.

The following proposition characterizes the proximity of two tokens.

Proposition 7. *The probability that among the set of tokens $\mathcal{T} = \{x_1, \dots, x_N\}$ with $\forall i = 1 \dots N$, $x_i \in \{0, \dots, c\}^n$ and x_i being a realization of a random variables X_i, there exists at least two tokens in the 2-ball of radius $m/2$, $m \in [\![0, c]\!]$, centered at the origin, is*

$$\begin{aligned} & P(\exists i \neq j \in \{1, \dots, N\},\ \|X_i - X_j\|_2 \le m) \\ & \qquad \le N(N-1) \sum_{k=0}^{m^2} [z^k] \left(G_{(U-V)^2}(z) \right)^n, \end{aligned}$$

where $G_{(U-V)^2}(z) = \sum_{k=0}^{c} P(|U - V| = k)\, z^{k^2}$ and the distribution of U, V is the same as the marginal distribution of X_i, X_j.

Proof. Computing, we have

$$\begin{aligned} & P(\exists i \neq j \in \{1, \dots, N\},\ \|X_i - X_j\|_2 \le m) \\ & \quad = P\left(\cup_{1 \le i \neq j \le N},\ \|X_i - X_j\|_2 \le m \right) \\ & \quad \le \sum_{1 \le i \neq j \le N} P\left(\|X_i - X_j\|_2 \le m \right) \\ & \quad = N(N-1)\, P(\|X_i - X_j\|_2 \le m), \\ & \quad = N(N-1) \sum_{k=0}^{m^2} [z^k] \left(G_{(U-V)^2}(z) \right)^n, \end{aligned} \tag{7}$$

using Proposition 6. □

It is clear that $\lim_{n \to \infty} P(\|X_i - X_j\| \le m) = 0$. To have an estimate on the convergence rate, we apply a central limit theorem as follows. For each couple (X_i, X_j), $1 \le i < j \le N$, we associate n scalar random variables A_{ijk}, $k = 1 \dots n$ defined by $A_{ijk} = (X_i^{(k)} - X_j^{(k)})^2$ and their probability law is given by (6). We denote by μ and $\sigma^2 \neq 0$ the expectation and the variance of A_{ijk} respectively. The central limit theorem asserts that $1/n\ \sum_k A_{ijk}$ converges in probability to $\mathcal{N}(\mu, \sigma^2/n)$, hence we get for $m \in [\![0, c\sqrt{n}]\!]$:

$$\begin{aligned} & \lim_{n \to +\infty} P\left(\frac{\|X_i - X_j\|_2}{\sqrt{n}} \le m \right) \\ & \quad = \lim_{n \to +\infty} P\left(\frac{\sum_{k=1}^{n} A_{ijk}}{n} \le m^2 \right) \\ & \quad = \int_{-\infty}^{m^2} \frac{1}{\sqrt{2\pi}\sigma/\sqrt{n}} \exp\left(-\frac{1}{2} \left(\frac{x - \mu}{\sigma/\sqrt{n}} \right)^2 \right) \mathrm{d}x \\ & \quad = \frac{1}{2} \left(1 + \operatorname{erf}\left(\zeta_n(m^2) \right) \right), \end{aligned} \tag{8}$$

where $\zeta_n(x) = \frac{x - \mu}{\sqrt{2}\sigma/\sqrt{n}}$ and erf is the *Gauss error function* defined by $\operatorname{erf}(x) = \dfrac{2}{\sqrt{\pi}} \displaystyle\int_0^x \exp(-t^2)\, \mathrm{d}t$. We have shown the following theorem, illustrated in Fig.2, which represents the probability $P(\|X_i - X_j\|_2^2 \le m^2)$, $m = 0 \dots \sqrt{n}c$, where X_i, X_j follow a uniform distribution on $[\![0, c]\!]^n$ ($c = 9, n = 256$).

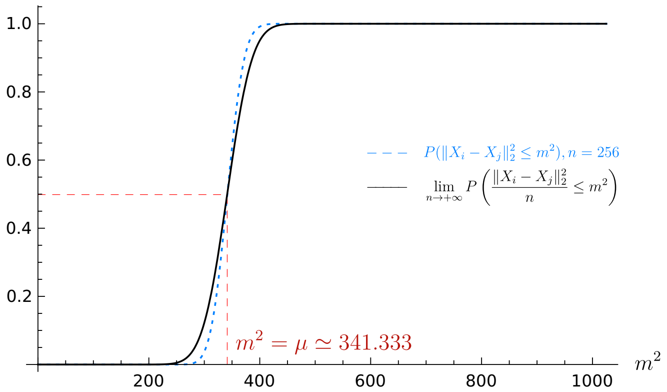

Fig. 2. *(dashed line)* Probability $P(\|X_i - X_j\|_2 \leq m)$ where X_i, X_j are uniform random variables valued in $[\![0, c]\!]^n$, $n = 256$, $c = 9$ computed using (3)-(4) for $m = 0, \ldots, n\,c$. μ corresponds to the expectation of $(X_i^{(k)} - X_j^{(k)})^2$. *(continuous line)* Limit case where $n \to +\infty$ computed using (8) (see Theorem 8).

Theorem 8. *Consider two random variables X_i, X_j valued in $\{0, \ldots, c\}^n$. The marginal distributions of their components are the same and we note the expectation μ. Then, we have*

$$\lim_{n\to+\infty} P\left(\frac{\|X_i - X_j\|_2}{\sqrt{n}} \leq m\right) = \begin{cases} 1 & \textit{if } m^2 > \mu \\ \frac{1}{2} & \textit{if } m^2 = \mu \\ 0 & \textit{if } m^2 < \mu \end{cases} .$$

Theorem 8 answers the question raised at the beginning of Section II-B: for any fixed N, m and c, we can find n such that the probability that the assumptions of Theorem 1 are satisfied is arbitrary close to 1.

Remark 9. *The random variables A_{ijk}, $k = 1 \ldots n$ may not follow the same probability law. In that case, we shall use a more generalized version, namely the Lindeberg central limit theorem [5].*

Remark 10. *The rate of convergence of $\lim_{n\to+\infty} P\left(\frac{\|X_i - X_j\|_2}{\sqrt{n}} \leq m\right)$ can be precised using the asymptotic development of the Gauss error function as $n \to +\infty$ which is*

$$\textit{erf}(\zeta_n(m^2)) = \begin{cases} -1 + O\left(\sqrt{n}\frac{\exp(-nk^2)}{k\sqrt{\pi}}\right) & \textit{if } m^2 < \mu \\ 1 + O\left(\sqrt{n}\frac{\exp(-nk^2)}{k\sqrt{\pi}}\right) & \textit{if } m^2 > \mu, \end{cases} \tag{9}$$

where $k = \frac{m^2 - \mu}{\sqrt{2\pi}}$. We obtain

$$P\left(\frac{\|X_i - X_j\|_2}{\sqrt{n}} \leq m\right) = O\left(\sqrt{n}\ e^{-nk^2}\right) \textit{ if } m^2 < \mu.$$

We illustrate this convergence in Figure 2.

c) Generalization for other distances: Theorem 8 can be adapted for the d-norm, $1 \leq d < +\infty$, since $\|x\|_d^d = \sum_{k=0}^n |x_k|^d$ is a sum of n scalar. Next, we present other methods to derive Theorem 8 for the 1-norm and the infinity norm in the uniform case.

1**-norm ,** X_i, X_j **uniform.** Even if we can adapt the method presented above for the 2-norm, we propose another method to compute the probability $P(\|X_i - X_j\|_1 \leq m)$. Denote by $C_{n,m}^*$ the number of tuples $(x_i, x_j) \in ([\![0, c]\!]^n)^2$ such that $\|x_i - x_j\|_1 * m$, where $* \in \{=, \leq\}$. Enumerating the $c+1$ cases where the vector $x_i - x_j$ has its last component equal to $0, 1, \ldots, c$, we get the following recurrence relation

$$C_{n,m}^{=} = \sum_{k=0}^{c} C_{n-1,m-k}^{=}\, \delta_k^{=}, \tag{10}$$

where $\delta_m^{=}$ is the cardinal of the set $\{(a, b) \in \{0, \ldots, c\}^2,\ |a - b| = m\}$ for $m \in \{0, \ldots, c\}$ and is equal to

$$\delta_m^{=} = \begin{cases} c+1 & \text{if } m = 0 \\ 2(c - m + 1) & \text{if } m > 0 \end{cases} .$$

We can then use dynamic programming to compute $C_{n,m}^{=}$ efficiently and from the equality $C_{n,m}^{\leq} = \sum_{i=0}^{m} C_{n,i}^{=}$, we can compute the probability $P(\|U - V\|_1 \leq m)$.

Infinity norm, X_i, X_j **uniform.** In this case, we can easily derive an analytical expression for $P(\|X_i - X_j\|_\infty \leq m)$. For $m \in \{0, \ldots, c\}$, let $\Delta_{n,m}^{\leq}$ be the cardinal of the set $\{(u, v) \in (\{0, \ldots, c\}^n)^2,\ \|u - v\|_\infty \leq m\}$. The cardinals δ_m^*, $* \in \{=, \leq\}$, of the sets $\{(a, b) \in \{0, \ldots, c\}^2,\ |a - b| * m\}$ for $m \in \{0, \ldots, c\}$ are

$$\begin{aligned} \delta_m^{=} &= \begin{cases} c+1 & \text{if } m = 0 \\ 2(c - m + 1) & \text{if } m > 0 \end{cases} \quad \text{and} \\ \delta_m^{\leq} &= \sum_{i=0}^{m} \delta_i^{=} = (c+1)(2m+1) - m(m+1). \end{aligned} \tag{11}$$

Hence, using the relation $\Delta_{n,m}^{\leq} = (\delta_m^{\leq})^n$, we obtain

$$P(\|X_i - X_j\|_\infty \leq m) = \frac{\Delta_{n,m}^{\leq}}{(c+1)^{2n}} = (1 - z_m)^n ,$$

where $z_m = (c - m)(c + 1 - m)/(c + 1)^2$.
Following the proof of Proposition 7, we deduce the proposition:

Proposition 11. *The probability that, among the set of tokens $\mathcal{T} = \{x_1, \ldots, x_N\}$, where $\forall i = 1 \ldots N$, $x_i \in [\![0, c]\!]^N$, there exists at least two tokens in a ball of radius $m/2$, $m \in [\![0, c]\!]$ for the infinity norm is*

$$\begin{aligned} P(\exists i \neq j \in \{1, \ldots, N\},\ &\|X_i - X_j\|_\infty \leq m) \\ &\leq \frac{N(N-1)}{2}\ (1 - z_m)^n, \end{aligned} \tag{12}$$

where $z_m = (c - m)(c + 1 - m)/(c + 1)^2$. Moreover for every $m < c$ and N fixed, this probability tends to 0 as $n \to +\infty$.

III. Numerical results

We present in this section numerical results of the implementation of Algorithm 1 and a naive NSS algorithm using the Python v3.8 and the Numpy computing package and executed

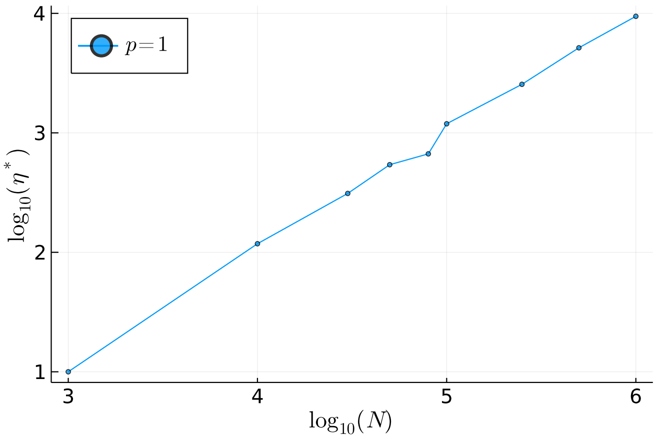

Fig. 3. Optimal value for the parameter η^* in Algorithm 1 for different values of N (hundred calls of Algorithm 1 were executed and the value of η^* is averaged). The parameter η^* is related to the size of the set I where the NNS is performed at the end of Algorithm 1, while N is the number of tokens and corresponds to the size of the set where the NNS is performed for the naive approach.

on the High Performance Computing cluster from Burgundy University[1].

The token set is generated from a discrete uniform distribution. We have $N = 10^6$ tokens of dimension $n = 256$ generated by a discrete uniform distribution valued in $[\![0, c]\!]$, $c = 9$. We denote the running times t_1 and t_n for Algorithm 1 and the naive NSS respectively. Recall the naive NSS consists in computing the minimal distance among $d(x_k, x_0)$, $k = 1 \ldots N$, hence its complexity is usually characterized by the number of calls of the distance function d. For algorithm 1, we choose $\eta = N/(n/p)$, which is an empirical value – different from the optimal value – and is incremented if x_1 is not the returned token. For $p = 1$, the average ratio t_n/t_1 over a hundred calls is more than 60 and the average optimal value η^* is represented in Fig.3 for several values of N.

IV. Conclusion

We present an algorithm based on projective filtering to compute the nearest neighbor of a token under some geometric assumptions considering different distances. We analyze the assumptions in terms of probability and show that, we need to choose high dimensional token (the dimension of a token being its number of digits) to satisfy the assumptions.

Other filtering functions can be used and an interesting perspective is to characterize them in terms of regularity and probability computation.

The proposed algorithm is exact if the parameter η is large enough, and this parameter can be estimated with respect to some refine conditions. In this direction, a scalability study shall be investigated together with average-case complexity for the presented algorithm.

[1] https://ccub.u-bourgogne.fr/dnum-ccub/

References

[1] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is “nearest neighbor” meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[2] Benjamin Bustos and Gonzalo Navarro. Probabilistic proximity searching algorithms based on compact partitions. *Journal of Discrete Algorithms*, 2(1):115–134, 2004.

[3] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM computing surveys (CSUR)*, 33(3):273–321, 2001.

[4] William Feller. *An introduction to probability theory and its applications. Vol. 1*. John Wiley & Sons, 1968.

[5] Jarl Waldemar Lindeberg. Eine neue herleitung des exponentialgesetzes in der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 15(1):211–225, 1922.

[6] Simon Schwerin. Blockchain and privacy protection in the case of the european general data protection regulation (gdpr): a delphi study. *The Journal of the British Blockchain Association*, 1(1):3554, 2018.

TABLE I
Nomenclature of parameters for Algorithm 1

Variable	**Description**	Order of magnitude
N	the number of token in the database	10^7
n	the dimension of a token space	100
c	the maximum value of a component of a token	10
p	the number of projections the number of tuples	$\leq n$
η	half of the size of the projected sets that are intersected to select potential neighbors	N/(n/p)