Austrian Journal of Statistics
February 2020, Volume 49, 18–30.
http://www.ajs.or.at/
doi:10.17713/ajs.v49i2.907

Fast Approximate Complete-data k-nearest-neighbor Estimation

Alejandro Murua
Département de mathématiques et de statistique
Université de Montréal
CP 6128, succ. centre-ville, Montréal
Québec H3C 3J7 Canada

Nicolas Wicker
Département de Mathématiques
Cité Scientifique
59655 Villeneuve d'Ascq
University of Lille

Abstract

We introduce a fast method to estimate the complete-data set of k-nearest-neighbors.This is equivalent to finding an estimate of the k-nearest-neighbor graph of the data. The method relies on random normal projections. The k-nearest-neighbors are estimated by sorting points in a number of random lines. For very large datasets, the method is quasi-linear in the data size. As an application, we show that the intrinsic dimension of a manifold can be reliably estimated from the estimated set of k-nearest-neighbors in time about two orders of magnitude faster than when using the exact set of k-nearest-neighbors.

Keywords: big data, dimension estimation, distance distribution, hash table, high dimensionality, symmetric k-nearest-neighbors.

1. Introduction

Let $\mathcal{D} = \{x_1, \ldots, x_n\} \subset \mathbf{R}^p$, $X \in \mathbf{R}^p$ be a point of interest. In this work we are concerned with the computation of the k-nearest-neighbors of X in $\mathcal{D}$. Let $\|\cdot\|$ denote the Euclidean norm in $\mathbf{R}^p$. Suppose that instead of computing all distances $\|X - x_i\|$, $i = 1, \ldots, n$, to obtain the exact k-nearest-neighbors of X, say k-NN(X) we are willing to sacrifice some precision and obtain an estimate of the set k-NN(X). The goal is to reduce the number of computations needed to find k-NN(X), provided that our estimate $\hat{\text{k}}$-NN(X) is sufficiently close to k-NN(X). Our method consists of projecting the data into a number, say m, of random projections so that we can recover k-NN(X) by searching for the k-nearest-neighbors in m one-dimensional subspaces. To measure closeness, we adopt the notion of ϵ-k-nearest-neighbors.

Definition 1.1. *ϵ-k-nearest-neighbor. We will say that a point y is an ϵ-k-nearest-neighbor of X if $\|y - X\| \leq (1 + \epsilon)\|x_{(k)} - X\|$, where $x_{(k)}$ is the k-nearest-neighbor of X.*

Different methods to estimate the ϵ-k-nearest-neighbors are available. Locality-sensitive hashing (Indyk and Motwani 1998) is one of the earliest methods suggested in the literature. It is highly popular. In this method, points are preprocessed and referenced by a hash-table so that a query point finds its approximate nearest-neighbor by querying the hash table. This step takes order $\mathcal{O}(pn^{1/\epsilon})$ operations. The method has been refined by Gionis, Indyk, and Motwani (1999) by improving the query time

to $\mathcal{O}(pn^{1/(1+\epsilon)})$ operations. Shakhnarovich, Darrell, and Indyk (2006) introduced randomness in the method, by estimating the hash function through random projections. Recently, Paulevé, Jégou, and Amsaleg (2010) used k-means to develop a highly competitive local-sensitivity hashing method.

All these methods are designed for the purpose of performing a query with a new external point. However, in this work we are rather concerned with the task of finding the k-nearest-neighbors for each point in $\mathcal{D}$. For example, we are interested in applying our methodology to construct the k-nearest-neighbor graph, a technique to built sparse graphs. For example, this graph is relevant when performing nonparametric regression (Altman 1992), intrinsic dimension estimation (Farahmand, Szepesvari, and Audibert 2007), dimension reduction (Bingham and Heikki 2001), outlier detection (Hautamäki, Kärkkäinen, and Fränti 2004), nonparametric clustering (Fränti, Virmajoki, and Hautamäki 2003), such as hierarchical clustering or superparamagnetic (Potts model) clustering (Murua, Stanberry, and Stuetzle 2008; Murua and Wicker 2014), for minimal spanning tree computation and density estimation (Stuetzle 2003), for semi-parametric Bayesian regression with autologistic distribution (Murua and Quintana 2017), and when performing Bayesian inference on complex data lying on a graph (Chekouo, Murua, and Raffelsberger 2015). It is clear that in these applications, nearest neighbors are searched for a dataset of size n, so applying the aforementioned methods would lead to a nearly quadratic time procedure in n.

The paper is organized as follows. Section 2 gives the basic definition and principles to search for k-NN using univariate projections. Section 2.3 introduces the main ideas and justifications for the method implemented in this work. An application to intrinsic dimension estimation is shown in Section 3. Section 4 contains some final thoughts about the procedure introduced in this work.

2. Method

Note that finding the nearest-neighbors in one-dimension is fairly trivial and fast if the data are already sorted, which can be done in $O(n \log n)$ operations. The basic idea is to search for the k-nearest-neighbors of the projection of X in each of the m subspaces. Let us denote by $\{x_{j(1)}, \ldots, x_{j(k)}\}$ the k-nearest-neighbors points to X in the j-th one-dimensional subspace, $j = 1, \ldots, m$. The search will produce at most $k \times m$ different points that are possible nearest-neighbors to X. The key idea is to search for the k-nearest-neighbors of X among these points instead of among the entire data set $\mathcal{D}$. This would yield our estimate set $\hat{\text{k}}\text{-NN}(X)$. Note that this task requires the computation of the true distances between X and the set of these points in $\mathbf{R}^p$, which requires at most $O(kmp)$ operations. This method will be effective only if (1) the estimate set $\hat{\text{k}}\text{-NN}(X)$ is reasonably close to the true set $\text{k-NN}(X)$, and (2) km is much smaller than n. In what follows, Section 2.1, we show this it is possible to accurately estimate the set of $\text{k-NN}(x)$ for all x in a given dataset by just looking at neighbors in one-dimensional projections. However, in order to make these ideas efficient, that is to convey low computational cost, we show in Section 2.3 that it is more effective to restrict the search of the $\text{k-NN}(x)$ to a sufficiently small ball centered at x and of constant radius (that is, independent of x). The method we have implemented, which is sketched in Algorithm 1, is based on this latter methodology.

2.1. Preliminary results

Suppose that m projections $a_1, \ldots, a_m$ are chosen at random from a Normal distribution $\text{N}(0, I_p)$. Suppose that

$$\|x_1 - X\| \leq \|x_2 - X\| \leq \cdots \leq \|x_k - X\| \leq \|x_i - X\|, \quad i \geq k.$$

Define $Y_{ij} = 1$ if $|a_j'(X - x_1)| < |a_j'(X - x_i)|$, $i \geq 2$, and let $Y_{ij} = 0$, otherwise. In order to correctly find the first-NN of X using only one projection, we would need to have all $Y_{ij} = 1$, $i \geq 2$. Let $\theta_{ij} = \text{Prob}(Y_{ij} = 1)$. Let $S_{im} = \sum_{j=1}^{m} Y_{ij}$. To correctly find the first-NN of X using m projections, we need $S_{im} > m/2$ for all $i \geq 2$. That is, we would like to have with very large

probability, say $1-\epsilon$ for some small value of ϵ,

$$\text{Prob}\left(S_{2m} > \frac{m}{2}, \ldots, S_{nm} > \frac{m}{2}\right) \geq 1-\epsilon.$$

In general, we also would like to correctly find k-NN(X). Let the event $E_{m(1)} = \{$ the first-NN of X is correctly identified using m random projections $\}$, and define the similar events $E_{m(\ell)} = \{$ the ℓ-nearest neighbor of X is correctly identified using m random projections $\}$, $\ell = 1, 2, \ldots, k$.

The following result says that we can get k-NN(X) with high accuracy provided that the number of projections is large enough and that the points are well separated.

Theorem 2.1. *Given $\epsilon > 0$, we have*

$$\text{Prob}\big(\cap_{\ell=1}^{k} E_{m(\ell)}\big) \geq 1-\epsilon.$$

provided that the number of random projections m satisfies

$$m \geq O\left(\log(n/\epsilon) + \log(k/4)\right). \tag{1}$$

This result is a special case of a more general result shown in Section 2.2. Its proof follow very similar steps as the proof of Theorem 2.3 below.

Algorithm 1 Method for finding the estimated k-nearest neighbors

Require: m, $\mathcal{D} = \{x_1, \ldots, x_n\}$ **(Preprocessing)**
  compute $\bar{x}$ the center of $x_1, \ldots, x_n$
  for all $j \in 1, \ldots, m$ **do**
    select a Gaussian random direction $a_j \sim \text{N}(0, I_p)$
    for all $i \in 1, \ldots, n$ **do**
      project each point $x_i - \bar{x}$ on the space generated by a_j producing coordinates w_{ij}
    end for
  end for
Require: the point of interest X **(Finding the k-nearest-neighbors of X)**
  for all $j \in 1, \ldots, m$ **do**
    project the point $X - \bar{x}$ on the space generated by a_j producing coordinates W_j
    compute the k-nearest-neighbors of W_j in the set $\{w_{ij}\}_{i=1}^{n}$
    map the k-nn of W_j to the original points, say $\{x_{j(1)}, \ldots, x_{j(k)}\} \subset \mathcal{D}$
  end for
  the estimated k-nearest neighbors of X are computed from the subset $\{x_{j(1)}, \ldots, x_{j(k)}\}_{j=1}^{m}$

The proof of Theorems 2.1 and 2.3 uses the following results.

Lemma 2.2. *Let $\delta > 0$, and let x, y be two arbitrary but different vectors in the sphere, S^{p-1} of $\mathbf{R}^p$, with p large. Let a be a p-variate standard normal random vector, i.e., $a \sim \text{N}_p(0, I_p)$. Then*

$$|\,\text{Cov}(a'x, a'y)| = |x'y| < \delta,$$

with probability larger than $1 - 4\exp(-p\delta^2/2)$. That is, in very high dimensions, the variables $a'x$ and $a'y$ are nearly independent.

Proof. Let $x^{\perp}$ be the orthogonal space of the vector x. Consider the equator $\mathcal{E} = x^{\perp} \cap S^{p-1}$ and the set near the equator $\mathcal{A}_\delta = \{w \in S^{p-1} : \min_{z\in\mathcal{E}} \|w - z\| < \delta\}$. Let $\mathcal{H}^+$ and $\mathcal{H}^-$ be the northern and southern hemispheres of the sphere with equator $\mathcal{E}$. Let $\lambda(\cdot)$ be the probability measure induced by the Lebesgue measure in the sphere S^{p-1}. According to the Theorem for the measure concentration for the sphere (Matoušek 2002, Section 14.1, pp. 330–332)

$$\lambda(\mathcal{A}_\delta) = \lambda(\{\mathcal{H}^+ \cup \mathcal{A}_\delta\}) + \lambda(\{\mathcal{H}^- \cup \mathcal{A}_\delta\}) - 1 \geq 1 - 4\exp(-p\,\delta^2/2).$$

Therefore, if $y \in \mathcal{A}_\delta$, and $z \in \mathcal{E}$ is such that $\|y - z\| \leq \delta$, we have

$$|\operatorname{Cov}(a'x, a'y)| = |x'y| = |x'(y - z)| \leq \|x\| \, \|y - z\| \leq \delta.$$

□

In words, this lemma says that for very large p, most of the points lie on the equator. Therefore, y lies most likely in $\mathcal{A}_\delta$, and consequently, the angle α between the vectors x and y is nearly a straight angle. This implies $\cos(\alpha) = x'y \approx 0$.

Proposition 1. *Let $\delta > 0$, and let $x, y, z \in R^p$, with p large. Suppose that $\|x - y\| < \|x - z\|$. Let a be a p-variate standard normal random vector, i.e., a is a random projection. Given $\epsilon > 0$, we have*

$$\operatorname{Prob}\big(|a'(x - y)|^2 \leq (1 + \epsilon)|a'(x - z)|^2\big) > 1/2 \tag{2}$$

with probability larger than $1 - 4\exp(-p\delta^2/2)$. In fact the probability increases very fast with the ratio $\|x - z\|^2/\|x - y\|^2$. See Figure 1 below.

Proof. Note that $\operatorname{Var}(a'(x - y)) = \operatorname{Var}((x - y)'a) = (x - y)' \operatorname{Var}(a)(x - y) = \|x - y\|^2$. Similarly $\operatorname{Var}(a'(x - z)) = \|x - z\|^2$. Hence, since a is normal, $a'(x - y) \sim \mathrm{N}(0, \|x - y\|^2)$, and $a'(x - z) \sim \mathrm{N}(0, \|x - z\|^2)$. Let $v = a'(x - y)/\|x - y\|$, $u = a'(x - z)/\|x - z\|$, and $r(x, y, z) = \|x - z\|/\|x - y\|$. We have

$$\begin{aligned}\operatorname{Prob}(|a'(x - y)|^2 \leq (1 + \epsilon)|a'(x - z)|^2) &= \operatorname{Prob}\big(v^2 \leq (1 + \epsilon)\, r^2(x, y, z)\, u^2\big) \\ &= \operatorname{Prob}\left(\left|\frac{v}{u}\right| \leq r(x, y, z)\sqrt{1 + \epsilon}\right).\end{aligned}$$

In the Appendix, we show that $w = u/v$ is distributed as a Cauchy distribution with density

$$f(w) = \left\{\pi\sqrt{1 - \rho^2}\left(1 + \big(\frac{w - \rho}{\sqrt{1 - \rho^2}}\big)^2\right)\right\}^{-1},$$

and distribution function $\arctan\big((w - \rho)/\sqrt{1 - \rho^2}\big)/\pi + 1/2$, with $\rho = \operatorname{Cov}(u, v) = (x - y)'(x - z)/(\|x - y\| \, \|x - z\|)$. Therefore

$$\operatorname{Prob}\left(\left|\frac{v}{u}\right| \leq r(x, y, z)\sqrt{1 + \epsilon}\right) = \\ \frac{1}{\pi}\arctan\left(\frac{r(x, y, z)\sqrt{1 + \epsilon} - \rho}{\sqrt{1 - \rho^2}}\right) + \frac{1}{\pi}\arctan\left(\frac{r(x, y, z)\sqrt{1 + \epsilon} + \rho}{\sqrt{1 - \rho^2}}\right).$$

Note that for any w and ρ, we have

$$\begin{aligned}\arctan\left(\frac{w}{\sqrt{1 - \rho^2}}\right) - \arctan\left(\frac{w - \rho}{\sqrt{1 - \rho^2}}\right) &= \int_{(w-\rho)/\sqrt{1-\rho^2}}^{w/\sqrt{1-\rho^2}} \frac{ds}{1 + s^2} \\ &\leq \frac{1}{1 + (w - \rho)^2/(1 - \rho^2)} \frac{\rho}{\sqrt{1 - \rho^2}} = \frac{\rho\sqrt{1 - \rho^2}}{1 + w^2 - 2\rho w}.\end{aligned}$$

Similarly,

$$\begin{aligned}\arctan\left(\frac{w + \rho}{\sqrt{1 - \rho^2}}\right) - \arctan\left(\frac{w}{\sqrt{1 - \rho^2}}\right) &= \int_{w/\sqrt{1-\rho^2}}^{(w+\rho)/\sqrt{1-\rho^2}} \frac{ds}{1 + s^2} \\ &\geq \frac{1}{1 + (w + \rho)^2/(1 - \rho^2)} \frac{\rho}{\sqrt{1 - \rho^2}} = \frac{\rho\sqrt{1 - \rho^2}}{1 + w^2 + 2\rho w}.\end{aligned}$$

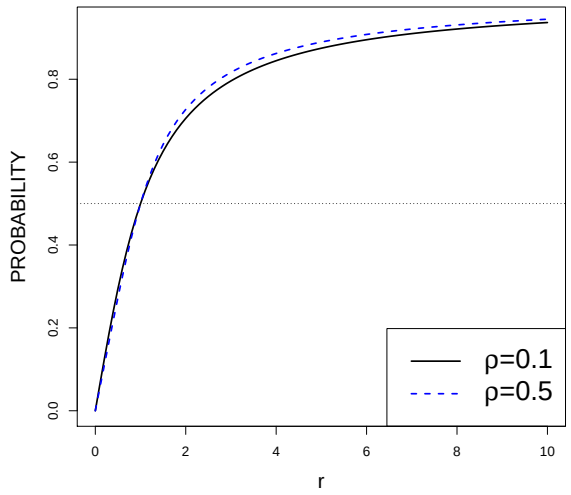

Figure 1: Probabilities (vertical axis) given by the Cauchy distribution. The horizontal axis is $r = \|x-z\|/\|x-y\|$.

Therefore, setting $w = r(x,y,z)\sqrt{1+\epsilon}$, we have

$$\pi \operatorname{Prob}\left(\left|\frac{v}{u}\right| \leq r(x,y,z)\sqrt{1+\epsilon}\right) \geq$$
$$2\arctan\left(\frac{r(x,y,z)\sqrt{1+\epsilon}}{\sqrt{1-\rho^2}}\right) + \frac{\rho\sqrt{1-\rho^2}}{1+w^2+2\rho w} - \frac{\rho\sqrt{1-\rho^2}}{1+w^2-2\rho w}$$
$$= 2\arctan\left(\frac{r(x,y,z)\sqrt{1+\epsilon}}{\sqrt{1-\rho^2}}\right) - \frac{4w\rho^2\sqrt{1-\rho^2}}{(1+w^2)^2-4\rho^2w^2}.$$

It is straightforward to show that for positive values of w, the function $g(w) = w/\{(1+w^2)^2 - 4\rho^2w^2\} = w/\{1+w^4+2w^2(1-2\rho^2)\}$ is maximized at $w = 3^{-1/2}[\sqrt{(1-2\rho^2)^2+3} - (1-2\rho^2)]^{1/2}$. Hence, for small values ρ, the function $g(w) \leq 1/\sqrt{3}$. Therefore, for small values of ρ,

$$\pi \operatorname{Prob}\left(\left|\frac{v}{u}\right| \leq r(x,y,z)\sqrt{1+\epsilon}\right) > 2\arctan\left(r(x,y,z)\sqrt{1+\epsilon}\right) - \frac{4}{\sqrt{3}}\rho^2\sqrt{1-\rho^2}.$$

By Lemma 2.2, on the event $\mathcal{A}_\delta$, we have $\rho \leq \delta$ with probability larger than $1 - 4\exp(-p\delta^2/2)$, In this event $4\rho^2\sqrt{1-\rho^2}/\sqrt{3}$ is of the order $\mathcal{O}(\delta^2)$. Now using the facts that $r(x,y,z)\sqrt{1+\epsilon} > 1$, that the function $\arctan(w)$ is strictly monotone, and that $2\arctan(1) = \pi/2$, we obtain the desired result for small enough values $\delta > 0$. □

2.2. A bound for the number of projections

Suppose that we have n points, $x_1, \ldots, x_n$. Let $a_1, \ldots, a_m \in \mathbf{R}^p$ be m random normal projections. We would like to correctly find k-NN(x_i), $i = 1, \ldots, n$. Let $\mathcal{N}_{i,t} = $ t-NN$(x_i) = \{x_{i(1)} \ldots, x_{i(t)}\}$, $t = 1, \ldots, k$. Let $\theta_{ij(\ell)} = \operatorname{Prob}(|a_j'(x_i - x_{i(\ell)})| < |a_j'(x_i - x_h)| : h \notin \mathcal{N}_\ell)$. For simplicity, let us assume that all $\theta_{ij(\ell)} = \theta$ for all i, j, ℓ. Let the event $E_{im(1)} = \{1\text{-NN}(x_i)$ is correctly identified using m random projections $\}$, and define the similar events $E_{im(\ell)} = \{$ the ℓ-nearest neighbor of x_i is correctly identified using m random projections $\}$, $\ell = 1, 2, \ldots, k$, and $i = 1, \ldots, n$. By Proposition 1, we may write $\theta = 1/2 + \delta$ for some $\delta \in (0, 1/2]$. The following result guarantees that for a sufficiently large number of projections, our algorithm solves the ϵ-k-nearest-neighbor problem.

Theorem 2.3. *Let $\delta = 1/2 - \theta$. Given $\epsilon > 0$, we have*

$$\text{Prob}\big(\cap_{i=1}^{n} \cap_{\ell=1}^{k} E_{im(\ell)}\big) \geq 1 - \epsilon.$$

provided that the number of random projections m satisfies

$$m \geq \Big(\frac{1}{2\delta^2} - 2\Big) O\Big(2\log(n/\epsilon) + \log(k/4)\Big). \tag{3}$$

Proof. First consider the case of the first-nearest-neighbor. Let $b(\theta) = (\theta - 1/2)/\sqrt{\theta(1-\theta)}$. Using the normal approximation to the Binomial distribution (note that S_m may be approximated by a Binomial(θ, m)), we need

$$\begin{aligned}\text{Prob}\big(S_{3m} > m/2, \ldots, S_{nm} > m/2\big) &= \text{Prob}\big(\min_{i\geq 3} Z_i > -\sqrt{m}\, b(\theta)\big) \\ &= 1 - \text{Prob}\big(\cup_{i=3}^{n}\big\{Z_i \leq -\sqrt{m}\, b(\theta)\big\}\big) \geq 1 - \sum_{i=3}^{n} \text{Prob}\big(Z_i \leq -\sqrt{m}\, b(\theta)\big) \\ &\geq 1 - n\,\text{Prob}\big(Z_1 \leq -\sqrt{m}\, b(\theta)\big) \approx 1 - n\,\Phi\big(-\sqrt{m}\, b(\theta)\big),\end{aligned}$$

where Z_i denotes a standardized Binomial(θ, m) random variable, and Φ is the distribution function of the standard normal. Now consider the general case. Using the Bonferroni inequality as above, we have

$$\text{Prob}\big(\cap_{i=1}^{n} \cap_{\ell=1}^{k} E_{im(\ell)}\big) \geq 1 - \sum_{i=1}^{n}\sum_{\ell=1}^{k} \text{Prob}(\bar{E}_{im(\ell)}),$$

where $\bar{E}_{im(\ell)}$ denotes the complementary event of $E_{im(\ell)}$, $i = 1, \ldots, n$ and $\ell = 1, \ldots, k$. Assuming that $k << n$, each one of the terms in this sum may be approximated by $n \times \Phi(-\sqrt{m}\, b(\theta))$. Hence,

$$\text{Prob}\big(\cap_{i=1}^{n} \cap_{\ell=1}^{k} E_{im(\ell)}\big) \geq 1 - kn^2 \times \Phi\big(-\sqrt{m}\, b(\theta)\big).$$

Therefore, we may ask that $kn^2 \times \Phi\big(-\sqrt{m} b(\theta)\big) \leq \epsilon$, which implies

$$-\Phi^{-1}\big(\epsilon/(kn^2)\big) \leq \sqrt{m}\, b(\theta).$$

Now, note that $\Phi^{-1}(p) = \sqrt{2}(\text{erf}^{-1})(2p-1)$, for $0 < p < 1$, where $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{t^2}\, dt$. In particular, one may use the approximation (Winitzki 2003, 2008)

$$\begin{aligned}\text{erf}^{-1}(z) \approx \text{sign}(z) \Bigg[&\Big\{ \big(2/a\pi + 1/2 \log(1-z^2)\big)^2 - 1/a \log(1-z^2) \Big\}^{1/2} \\ &- \big(2/a\pi + 1/2 \log(1-z^2)\big) \Bigg]^{1/2},\end{aligned}$$

where the constant $a = 0.147$. For very small $z \in (-1, 0)$, one can easily show that the above expression is approximately $-\sqrt{|\log(1-z^2)|}$. This is the case of interest for us, since we need to evaluate this function at $z = 2\epsilon/(kn^2) - 1$. Note as well that $\log(1-z^2) = \log(\frac{4\epsilon}{kn^2}(1 - \frac{\epsilon}{kn^2})) \approx= -\log(kn^2/(4\epsilon))$. Using these approximations, we obtain

$$m \geq \frac{2}{b^2(\theta)} O\Big(\log\Big(\frac{kn^2}{4\epsilon}\Big)\Big),$$

and hence (3). □

Note that even though we just need m of $O(\log(n/\epsilon))$, the constant depending on θ may dominate. For example for $\theta = 1/2 + \delta$, the term multiplying $O(\log(n/\epsilon))$ is δ^{-2}. So in practice we may need a large $m = O(\delta^{-2}\log(n/\epsilon))$. Figure 2 displays the number of necessary projections as function of the data size and the probability θ.

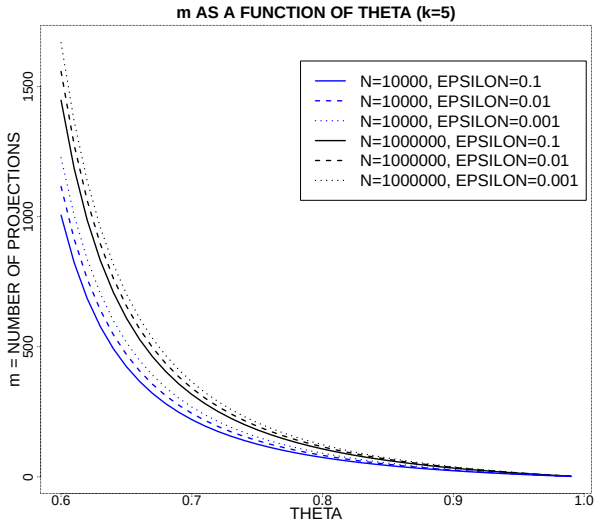

Figure 2: Number of projections as a function of the probability θ. Here we have used $k = 5$.

2.3. k-NN enclosed in a ball

Note that the computations above based on quantities such as S_m involved $O(n)$ one-dimensional comparisons for each x_i. Although this is much faster than $O(n)$ p-dimensional comparisons for each x_i when p is very large, it will be much faster and use far less memory if the computations to find the k-NN of the points where reduced to a few comparisons. This is the goal of this section. For example, for each point x_i and on each projection axis a_j, it may be enough to just look at the univariate $2k$-symmetric nearest-neighbors (sNN, for short) of $a_j'x_i$. The idea is to compute the estimated k-NN(x_i) from its m $2k$-sNN. With this method the estimated k-NN of the whole set of n points can be realized in order $\mathcal{O}(nm[\log(n) + k\log(km) + kp])$ as follows:

(i) Compute the m projections in $\mathcal{O}(nm)$ operations;

(ii) Sort the projected data in each line (projection) in $\mathcal{O}(mn\log(n))$ operations;

(iii) Take the $2k$-sNN for each x_i on each projection. Note that this is done very fast since the projected data is already sorted. This takes $\mathcal{O}(mnk)$ operations;

(iv) Compute all real distances between x_i and its m $2k$-sNN. This takes $\mathcal{O}(nmkp)$ operations;

(v) For each x_i, find its estimated k-NN from its m $2k$-sNN. This is done in $\mathcal{O}(nmk\log(mk))$ operations.

Let $a_1, \ldots, a_m$ be m random projections drawn from a Normal distribution $\mathrm{N}(0, I_p)$. An alternative way of realizing the same idea is to replace the $2k$-sNN for a subset of points that fall close to the target point. Let $z > 0$. For each x_i, define the subset $\mathcal{S}_i(z) = \{x_h : |a_j'(x_i - x_h)| < z$ for some $j = 1, \ldots, m\}$. For the method to work well, it suffices that all the $\mathcal{S}_i(z)$ contain about $2k$ points. So the parameter z must be chosen to ensure this with high probability. For a given projection a, consider the event $A_i(z) = \{x : |a'(x_i - x)| \leq z\}$. The distribution of the number of points $x_h \in D$, whose projections fall in the interval $[-z, z]$ is Binomial with parameters n and probability of success $\mathrm{Prob}(A_i(z))$. The expected number of points whose projections fall in this interval is $n\,\mathrm{Prob}(A_i(z))$. Equaling this to the desired number $2k$, we get that z must satisfy $\mathrm{Prob}(A_i(z)) = 2k/n$. The next result calculates this probability.

Theorem 2.4. *Suppose that the vector $a \sim \mathrm{N}(0, I_p)$ and x is distributed with density $f_p(x)$. Let $F_p(z) = \int_x \Phi(\|x\|^{-1}z)\, f_p(x)\, dx$, then $z = F_p^{-1}\big(\big[\frac{k}{n} + \frac{1}{2}\big]\big)$ satisfies $\mathrm{Prob}(|a'x| \leq z) = 2k/n$.*

In particular, if x is distributed uniformly in a ball of radius R, then

$$z = G_p^{-1}\Big(\frac{\pi^{p/2}R^p}{p\Gamma(1 + p/2)}\big[\frac{k}{n} + \frac{1}{2}\big]\Big),$$

with $G_p(z) = \int_0^R \Phi(s^{-1}z)\, s^{p-1} ds$.

Proof. Note that

$$\begin{aligned}\text{Prob}(|a'x| \le z) &= \int_x \int_{-z}^{z} p(v = a'x, x)\, dv dx = \int_x \int_{-z}^{z} p(v|x) f_p(x)\, dv dx \\ &= \int_x \big(2\Phi(z/\|x\|) - 1\big) f_p(x)\, dx = 2\int_x \Phi\big(z/\|x\|\big) f_p(x)\, dx - 1\end{aligned}$$

If x is distributed uniformly in a ball of radius R, then

$$\text{Prob}(|a'x| \le z) = \frac{2\Gamma(1 + p/2)}{\pi^{p/2}} pR^{-p} \int_0^R \Phi(s^{-1}z)\, s^{p-1} ds - 1.$$

□

This theorem gives the value of z. The next theorem establishes that our algorithm finds the k-NN of the entire dataset with high probability. Let $x_{i(1)}, \ldots, x_{i(k)}$ be the k-NN of the point x_i, $i = 1, \ldots, n$. Assume that $\max_{i,\ell}\{\|x_i - x_{i(\ell)}\|\} \le r$. Define the events $A^j_{i,\ell}(z) = \{|a'_j(x_i - x_{i(\ell)})| \le z\}$, $\ell = 1, \ldots, k$.

Theorem 2.5. *For any* $\epsilon > 0$,

$$\text{Prob}\big(\cap_{i=1}^n \cap_{\ell=1}^k \cup_{j=1}^m A^j_{i,\ell}(z)\big) \ge 1 - \epsilon,$$

provided that $m \ge \log(nk/\epsilon)/\log(2\Phi(z/r) - 1)$.

Proof: Recall that $a'_j(x_i - x_{i(\ell)}) \sim \text{N}(0, \|x_i - x_{i(\ell)}\|^2)$. So that $\text{Prob}(A^j_{i,\ell}(z)) = 2\Phi(z/\|x_i - x_{i(\ell)}\|) - 1 \ge 2\Phi(z/r) - 1$, for all $\ell = 1, \ldots, k$, and $i = 1, \ldots, n$. We have

$$\begin{aligned}\text{Prob}\Big(\cap_{i=1}^n \cap_{\ell=1}^k \cup_{j=1}^m A^j_{i,\ell}(z)\Big) &= 1 - \text{Prob}\Big(\cup_{i=1}^n \cup_{\ell=1}^k \cap_{j=1}^m (A^j_{i,\ell}(z))^c\Big) \\ &\ge 1 - \sum_{i=1}^n \sum_{\ell=1}^k \prod_{j=1}^m \Big(1 - \text{Prob}(A^j_{i,\ell}(z))\Big) \\ &= 1 - \sum_{i=1}^n \sum_{\ell=1}^k \prod_{j=1}^m 2\Big(1 - \Phi\big(z/\|x_i - x_{i(\ell)}\|\big)\Big) \\ &\ge 1 - nk2^m(1 - \Phi(z/r))^m.\end{aligned}$$

Therefore, to show the theorem, it is sufficient to ask $nk2^m(1 - \Phi(z/r))^m \le \epsilon$, which implies $m \ge \log(\epsilon/(nk))/\log(2(1 - \Phi(z/r)))$.

How large is r? To answer this question, we need to know the distribution of the data (at least locally). We consider the same setup introduced in the work of von Luxburg, Radl, and Hein (2014). For any $\epsilon > 0$, and $x \in \mathbb{R}^p$, let $B_\epsilon(x)$ be the ball of radius ϵ centered at x. Suppose that the data $\mathcal{D} \subset \mathbb{R}^p$ is such that $\text{support}(\mathcal{D})$ verifies that for all x in its boundary, $\text{vol}(B_\epsilon(x) \cap \partial\text{support}(\mathcal{D})) \ge \alpha\text{vol}(B_\epsilon(x))$, for some $0 < \alpha \le 1$, where $\partial\text{support}(\mathcal{D})$ denotes the boundary of $\text{support}(\mathcal{D})$. Proposition 30 in (von Luxburg *et al.* 2014) on the maximum distance between a point x_i and its nearest k-th neighbor, $x_{i(k)}$ states that:

$$\text{Prob}\left(\max_i \|x_i - x_{i(k)}\| \ge s\left(\frac{k}{n}\right)^{1/p}\right) \le n \exp\{-k/12\}$$

where $f_{\min} > 0$ is the minimum density in $\mathcal{D}$, $s = 2/(f_{\min}\text{vol}(B_1(0))\alpha)^{1/p}$. Assuming that the data were generated with a uniform distribution, we have $f_{\min} = [\text{vol}(\text{support}(\mathcal{D}))]^{-1}$. Concerning, α,

we can easily bound it, if we assume that $\mathrm{support}(\mathcal{D})$ is a volume without spikes. For instance, if $\mathrm{support}(\mathcal{D})$ is the unit ball, α can be made arbitrarily close to 0.5 (Li 2011). Therefore, if $\mathrm{support}(\mathcal{D})$ is a ball of radius R, we may take $r = 2R\left(\frac{2k}{n}\right)^{1/p}$, provided that $k \geq \log n$, and n is large.

3. Numerical application

As the algorithm only retrieves approximately the k-nearest neighbors of a set of points, it is important to show that the obtained points are nevertheless useful for practical applications. The application we have chosen is intrinsic dimension estimation. Different definitions can be given of the intrinsic dimension which can be found in Kegl (2002), informally in the folklore it refers to the useful number of parameters needed to describe the data. Various methods has been proposed, the correlation dimension estimation (Grassberger and I. Procaccia 1983), the capacity dimension (Kegl 2002) and the maximum likelihood dimension estimation (Levina and Bickel 2004; Bouveyron, Celeux, and Girard 2011) among others. Recently, Farahmand *et al.* (2007) have proposed the following estimator of the intrinsic dimension of a manifold at at point x

$$\hat{d}(x) = \log 2 / \log\big(\hat{r}_k(x)/\hat{r}_{\lceil k/2 \rceil}\big)$$

where $\hat{r}_\ell(x)$ represents the distance of x to its ℓ-th nearest neighbor. To estimate the dimension of the whole manifold, we simply compute the empirical mean of it : $\hat{d} = \frac{1}{n}\sum_{i=1}^{n} \hat{d}(x_i)$. Using this estimator, we compare the estimation of the intrinsic dimension obtained by (a) using the exact k-nearest neighbors, and (b) the approximate k-nearest neighbors yielded by our procedure. We compared the estimators on several artificially generated datasets. These consisted of data drawn from a uniform distribution over $B_1(0)$, the unit ball in $\mathbb{R}^p$ with $p \in \{5, 10, 20, 50\}$. The dataset sizes were set to 10^5 or 10^6. The results, for the number of nearest neighbors equals to $k = 10$ or 30, and the number of random projections equals to 10, 20 or 100, are displayed in Table 1. The table shows the dimension estimations as well as the execution times.

Although the results are comparable in terms of quality, the execution times are largely in favor of our approximate method; particularly, when the datasets contain a million points. As the number of projection increases, the results yielded by the approximate k-nearest-neighbors method get closer to the estimation given by the exact k-nearest-neighbor method. Also, when the number of neighbors increases from 10 to 30, the results associated with the approximate k-nearest-neighbors method improve with an increase in the number of projections. The best results for our approximate k-nearest-neighbors algorithm are observed for 1 million point datasets, 10 neighbors and 100 projections.

The approximate k-nearest-neighbors method tends to underestimate the true dimension. A possible explanation for this behaviour, is that the farther from a given point x we go, the less accurate the estimation of its true ℓ-nearest neighbor, for large $\ell \leq k$. This, in turn, has an effect on the formula for estimation of the dimension. More specifically, there is a trailing error in the consecutive estimation of the k-nearest-neighbors. Therefore, the error in the estimation of say, r_k in the formula of the dimension tends to be much larger than the error in the estimation of $r_{\lceil k/2 \rceil}$. This would produce estimated ratios $\hat{r}_k/\hat{r}_{\lceil k/2 \rceil} > r_k/r_{\lceil k/2 \rceil}$. Consequently, the approximate k-nearest-neighbors estimate of the dimension $\hat{d}$ tends to be smaller than the estimate of d made with the exact k-nearest-neighbors. More specifically, let $\hat{r}_k = r_k + e_k$, and $\hat{r}_{\lceil k/2 \rceil} = r_{\lceil k/2 \rceil} + e_{\lceil k/2 \rceil}$. Suppose that $e_k = e_{\lceil k/2 \rceil} + \nu_k$, with $\nu_k > 0$. Then the dimension will be underestimated if $\hat{r}_k/\hat{r}_{\lceil k/2 \rceil} > r_k/r_{\lceil k/2 \rceil}$. This holds if and only if $e_k/e_{\lceil k/2 \rceil} > r_k/r_{\lceil k/2 \rceil}$. Equivalently, one would need to have $\nu_k/e_{\lceil k/2 \rceil} > (r_k/r_{\lceil k/2 \rceil}) - 1$ for the dimension to be underestimated. This simple computation shows that underestimation may be absent for well-separated (spread) data.

We have also tested our method on two different datasets for which we have a pretty good estimation of the "true" intrinsic dimension. The first dataset is the MNIST (LeCun, Bottou, Bengio, and Haffner 1998) which consists in 60000 black and white images of size 28×28. Usually, the intrinsic dimension deemed to be about 14 (Hein and Audibert 2014; Costa and Hero III 2006; Facco, d'Errico, Rodriguez,

and Laio 2017). Here, with $k = 30$ neighbors and $m = 100$ projections we estimate the intrinsic dimension to be 13.8 Using the exact k-neighborhood, the intrinsic dimension is estimated to be 16.2. The computation of the approximate procedure took only 298 seconds, while the one that uses the exact k-neighborhood took $3,020$ seconds, that is, more than ten times longer than the approximate procedure.

The second dataset is the Dota 2 dataset collected on the UCI repository dataset website (Dheeru and Karra Taniskidou 2017). Dota 2 contains $102,944$ Dota 2 computer game results. The games are played by two teams of 5 players each. Each player can choose a hero among 113 possible heroes. Given that a hero can only be chosen by one player, the 113-dimensional rows contain only 10 values different from 0: -1 for one team and 1 for the other team; the value 0 is given to heroes that have not been chosen by any of the teams. A reasonable guess for the intrinsic dimension is 10. In practice, we got the estimates 7.5 with the exact k-nearest neighbors algorithm in $1,221$ seconds, and 9.3 with the approximate k-neighborhood, in 83 seconds, for $k = 10$ and $m = 100$

Size	dim	k	dimension exact k-NN	timeE (seconds)	dimension approx. k-NN Number proj.			timeA (seconds) Number proj.		
					10	20	100	10	20	100
100K	5	10	8.2	166	5.5	5.8	6.4	6	12	59
100K	10	10	15.6	248	10.8	11.5	12.7	6	12	60
100K	20	10	28.7	355	18.9	20.4	23.2	6	13	62
100K	50	10	58.4	686	35.4	39.2	45.9	8	15	67
100K	5	30	5.6	322	3.8	4.0	4.3	18	40	221
100K	10	30	10.5	404	7.5	7.9	8.7	18	41	235
100K	20	30	19.2	498	13.0	14.1	15.9	19	42	222
100K	50	30	38.7	810	24.4	26.9	31.6	20	44	232
1M	5	10	8.3	22198	5.5	5.8	6.3	63	136	716
1M	10	10	16.0	30690	10.8	11.5	12.5	66	141	721
1M	20	10	29.7	41546	18.9	20.4	23.0	70	147	736
1M	50	10	62.4	66522	35.3	39.2	45.8	84	165	796
1M	5	30	5.6	32862	3.8	4.0	4.3	188	444	2635
1M	10	30	10.8	39273	7.5	7.9	8.6	190	491	2784
1M	20	30	20.0	49566	13.0	14.1	15.8	197	467	2789
1M	50	30	41.6	79428	24.4	26.9	31.4	218	495	2940

Table 1: Intrinsic dimension estimation results obtained using exact, random and approximate k-nearest neighbors; timeE and timeA stand, respectively, for the exact and the approximate k-nearest neighbors execution times.

4. Conclusion

An efficient algorithm for computing approximate k-nearest neighbors has been introduced in this work. The method is easy to implement. It essentially comes down to performing random projections, sorting points along these projection lines, and computing the distances to the nearest neighbors found along these lines.

We have shown its efficiency with an application to dimension estimation based on k-nearest neighbors. Dimensions have been estimated on artificial as well as real examples, such as the MNIST and the Dota 2 games dataset which consists of more than 1 million observations. Our experiments show that we still obtain very good estimates of the intrinsic dimensions while at the same time we decrease the computing time to about two orders of magnitude. A potential improvement in performance could be achieved by considering low-discrepancy sequences instead of random projections. Low-discrepancy sequences, also known as quasi-random sequences, are such that any considered subsequence has a distribution close to the uniform distribution. In our case, we would be interested in using low-discrepancy sequences on the hypersphere (Wong, Luk, and Hen 1997). The idea is to

explore the space of directions in a quasi-random way aiming at exploring it in a more systematic way than just randomly. This is left for future work.

Acknowledgments

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) through Discovery grant number 2019-05444, and also by the Labex CEMPI (ANR-11-LABX-0007-01).

Appendix

Let $b = a'(x-y)/\|x-y\|$, and $d = a'(x-z)/\|x-z\|$, where $a \sim N_p(0, I_p)$, and $x, y, z \in R^p$. In this section we show that the distribution of $U = b/d$ is a Cauchy distribution. First, note that every linear combination of b and d, say $\alpha_1 b + \alpha_2 d = a'(\alpha_1(x-y) + \alpha_2(x-z))$ is a Normal random variable. Hence, (b, d) is a bivariate normal vector. Its mean is zero, and its variance-covariance matrix is $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$, with $\rho = (x-y)'(x-z)/(\|x-y\|\,\|x-z\|)$. Let $(U, V) = (b/d, d)$. By the change of variable theorem, the density of (U, V) is given by

$$f_{(U,V)}(u,v) = f_{(b,d)}(uv, v) \begin{vmatrix} v & u \\ 0 & 1 \end{vmatrix} = \frac{|v|}{2\pi\sqrt{1-\rho^2}} \exp\left\{ -\frac{v^2}{2(1-\rho^2)} \left(u^2 - 2\rho u + 1\right) \right\}.$$

Note that $u^2 - 2\rho u + 1 \geq 1 - \rho^2$ is always strictly positive. Let $\sigma^2 = (1-\rho^2)/(u^2 - 2\rho u + 1)$. Our goal is to find $f_U(u)$. This is given by

$$\begin{aligned} f_U(u) &= \int_v \frac{|v|}{2\pi\sqrt{1-\rho^2}} \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{v^2}{2\sigma^2}\}\, dv \times \sigma\sqrt{2\pi} \\ &== \frac{2\sigma}{\sqrt{2\pi(1-\rho^2)}} \int_0^{+\infty} \frac{v}{\sigma\sqrt{2\pi}} \exp\{-\frac{v^2}{2\sigma^2}\}\, dv \\ &= \sqrt{\frac{2}{\pi}} \frac{\sigma^2}{\sqrt{1-\rho^2}} \int_0^{+\infty} \frac{v}{\sqrt{2\pi}} \exp\{-\frac{v^2}{2}\}\, dv \\ &= \frac{\sigma^2}{\pi\sqrt{1-\rho^2}} = \frac{\sqrt{1-\rho^2}}{\pi(u^2 - 2\rho u + 1)} \\ &= \frac{1}{\pi\sqrt{1-\rho^2}} \left(1 + \left(\frac{u-\rho}{\sqrt{1-\rho^2}} \right)^2 \right)^{-1}, \end{aligned}$$

which is the Cauchy density with location ρ and scale $\sqrt{1-\rho^2}$.

References

Altman NS (1992). "An Introduction to Kernel and Nearest-neighbor Nonparametric Regression." *The American Statistician*, **46**(3), 175–185.

Bingham E, Heikki M (2001). "Random Projection in Dimensionality Reduction: Applications to Image and Text Data." In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*.

Bouveyron C, Celeux G, Girard S (2011). "Intrinsic Dimension Estimation by Maximum Likelihood in Isotropic Probabilistic PCA." *Pattern Recognition Letters*, **32**(14), 1706–1713.

Chekouo T, Murua A, Raffelsberger W (2015). "The Gibbs-plaid Biclustering Model." *Annals of Applied Statistics*, **9**(3), 1643–1670.

Costa JA, Hero III AO (2006). "Determining Intrinsic Dimension and Entropy of High-dimensional Shape Spaces." *Statistics and Analysis of Shapes*, p. 231âĂŞ252.

Dheeru D, Karra Taniskidou E (2017). "UCI Machine Learning Repository." URL http://archive.ics.uci.edu/ml.

Facco E, d'Errico M, Rodriguez A, Laio A (2017). "Estimating the Intrinsic Dimension of Datasets by a Minimal Neighborhood Information." *Scientific Reports*, **7**(12140).

Farahmand AM, Szepesvari C, Audibert JY (2007). "Manifold-adaptive Dimension Estimation." In *Proceedings of ICML-2007*.

Fränti P, Virmajoki O, Hautamäki V (2003). "Graph-based Agglomerative Clustering." In *Proceedings of the third IEEE Int. Conf. on Data Mining*, pp. 525–528.

Gionis A, Indyk P, Motwani R (1999). "Similarity Search in High Dimensions via Hashing." In *Proceedings of the 25th VLDB conference*.

Grassberger P, I Procaccia I (1983). "Measuring the Strangeness of Strange Attractors." *Physica D: Nonlinear Phenomena*, **9**(1), 189–208.

Hautamäki V, Kärkkäinen I, Fränti P (2004). "Outlier Detection Using k-nearest Neighbour Graph." In *Proceedings of the 17th International Conference on Pattern Recognition*.

Hein M, Audibert JY (2014). "Intrinsic dimensionality estimation of submanifolds in $\mathbb{R}^d$." In *Proceedings of the 22nd international conference on Machine learning*, pp. 289–296.

Indyk P, Motwani R (1998). "Approximate Nearest Neighbor towards Removing the Curse of Dimensionality." In *Proceedings of the 30th symposium on theory of computing*, pp. 604–613.

Kegl B (2002). "Intrinsic Dimension Estimation Using Packing Numbers." In *Advances in Neural Information Processing Systems*.

LeCun Y, Bottou L, Bengio Y, Haffner P (1998). "Gradient-based Learning Applied to Document Recognition." In *Proceedings of the IEEE*, volume 86, pp. 2278–2324.

Levina E, Bickel PJ (2004). "Maximum Likelihood Estimation of Intrinsic Dimension." In *Proceedings of the 2004 Conference, NIPS*.

Li S (2011). "Concise Formulas for the Area and Volume of a Hyperspherical Cap." *Asian Journal of Mathematics and Statistics*, **4**(1), 66–70.

Matoušek J (2002). *Lectures On Discrete Geometry*. Springer-Verlag, New York.

Murua A, Quintana F (2017). "Semiparametric Bayesian Regression via Potts Model." *Journal of Computational and Graphical Statistics*, **26**(2), 265–274.

Murua A, Stanberry L, Stuetzle W (2008). "On Potts Model Clustering, Kernel K-Means and Density Estimation." *Journal of Computational and Graphical Statistics*, **17**(3), 629–658.

Murua A, Wicker N (2014). "The Conditional-Potts Clustering Model." *Journal of Computational and Graphical Statistics*, **23**, 717–739.

Paulevé L, Jégou H, Amsaleg L (2010). "Locality Sensitive Hashing: A Comparison of Hash Function Types and Querying Mechanisms." *Pattern Recognition Letters*, **31**(11), 1348–1358.

Shakhnarovich G, Darrell T, Indyk P (2006). *Nearest-neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press. Chapter 3.

Stuetzle W (2003). "Estimating the Cluster Tree of a Density by Analyzing the Minimal Spanning Tree of a Sample." *Journal of Classification*, **20**(1), 25–47.

von Luxburg U, Radl A, Hein M (2014). "Hitting Commute Times in Large Random Neighborhood Graphs." *Journal of Machine Learning Research*, **15**, 1751–1798.

Winitzki S (2003). "Uniform Approximations for Transcendental Functions." In *Proceedings of ICCSA-2003*, volume LNCS 2667/2003, p. 962.

Winitzki S (2008). "A Handy Approximation for the Error Function and Its Inverse." Lecture notes, URL http://sites.google.com/site/winitzki/sergei-winitzkis-files/erf-approx.pdf.

Wong TT, Luk WS, Hen PA (1997). "Sampling with Halton Points on n-Sphere." *Journal of Graphics Tools*, **2**(2), 9–24.

Affiliation:

Alejandro Murua
Université de Montréal
CP 6128, succ. centre-ville, Montréal
Québec H3C 3J7 Canada
E-mail: murua@dms.umontreal.ca
URL: https://dms.umontreal.ca/~murua/