Fast and secure similarity search in high dimensional space

Teddy Furon, Hervé Jégou, Laurent Amsaleg, Benjamin Mathon

Fast and secure similarity search in high dimensional space

Teddy Furon, Hervé Jégou, Laurent Amsaleg, and Benjamin Mathon

Inria Rennes, Campus de Beaulieu, Rennes, France

***Abstract*— Similarity search in high dimensional space database is split into two worlds: i) fast, scalable, and approximate search algorithms which are not secure, and ii) search protocols based on secure computation which are not scalable. This paper presents a one-way privacy protocol that lies in between these two worlds. Approximate metrics for the cosine similarity allows speed. Elements of large random matrix theory provides security evidences if the size of the database is not too big with respect to the space dimension.**

I. Introduction

Quickly identifying in a very large database the elements that are the most similar to a given query is a central need for many applications. While this has been solved many years ago for traditional relational databases, similarity search in high-dimensional space is still challenging. During the last decade, the quest for scalability was paramount. This is particularly difficult because the curse of dimensionality severely hurts the performance of retrieval algorithms. The most efficient solutions run *approximate* searches, where efficiency is traded-off against a reduction of the search quality. State-of-the-art retrieval techniques now cope with databases comprising millions to billions elements, return answers very fast, and the quality of the results is appropriate in most applications.

A. The need for security and privacy

Recently, other challenges have raised in this field: security and privacy. This is today critical as the trend is to *outsource* to a third party data, processing or both. Outsourcing is beneficial as one might not have enough storage capacity and/or computing power and/or the capacity to enforce 24×7 availability of services. But it raises security and privacy problems. Typically, the Owner of the database subcontracts the search task to a Server. This actor is not fully trusted, more specifically it is assumed to be honest but curious: It may infer information about the database or the queries of the Client.

Outsourcing challenges biometric identification. The main axiom in biometric claims that no database can be stored securely [11]. Therefore, the Server cannot store the database of biometric templates in the clear (*i.e.* not protected) since a pirate would steal these highly sensitive data. In the same way, the User is reluctant in sending biometric template query in the clear. Similarity search is also the cornerstone of some classification algorithms. A class is associated to each vector of the database, and the goal is to predict the class of the query vector from the class of its most similar vectors in the database. The Owner does not want to share his database as this is the fruit of his know-how in collecting and assessing the quality of these data. The Client is interested in the prediction value but does not want to disclose his query vector for some privacy issues. This happens in applications such as medical diagnostic (vectors are features extracted from medical records) or user recommendation system (vectors are user profiles).

WIFS'2013, November 18-21, 2013, Guangzhou, China.
ISBN 978-1-4673-5593-3
Supported by french project Secular ANR-12-CORD-0014.

B. The past approaches

Security and privacy in similarity search has become a hot topic in the field of information security. A special issue of the IEEE Signal Processing Magazine has been recently published [1]. It is striking that all its articles propose solutions based on homomorphic cryptography (mostly with the Pallier cryptosystem). An even more recent article pushes this state-of-the-art further using Gentry's fully homomorphic cryptosystem [11]. This common approach has been coined 'Signal processing in the encrypted domain': The encrypted similarity metric is computed from encrypted vectors, so that the Server neither sees the data nor their similarities. The security and privacy inherit from the computational security of these cryptosystems. They are semantically secure in the sense that nothing about the plaintext, be it vector or metric, can be inferred from the ciphertext. They are also robust to known plaintext attacks: The secret key cannot be disclosed when the attacker observes pairs of plaintext / ciphertext.

C. Our contributions

Our approach is motivated by the facts that encryption, homomorphic operations, and decryption are time-consuming. Ciphertexts are also big and consume a lot of storage and bandwidth. Some efforts have been made to combat these pitfalls. However, computation runtime and bandwidth remain bigger by several orders of magnitude than the state-of-the-art approximate similarity search when security is not enforced.

So far, the trade-off between security and scalability seems to be exclusive: Solutions are either very secure or very

scalable. This paper aims at designing a solution striking a softer trade-off. Our keystone idea is to enforce security with signal processing tools. We also resort to cryptography, but only to symmetric encryption, which is much faster than homomorphic schemes. Our strategy is to start from a scalable search algorithm and to modify it to gain security. This is achieved by requiring more complexity on the Client side, but far less than protocols based on homomorphic encryption do.

We stress three main contributions:

- Approximated metrics enabling a fast search dedicated to the cosine similarity (Section II-B);
- A protocol enabling both one-way privacy / security and speed (Section III);
- A security assessment based on theoretical elements of statistics of high dimensional signals, and especially the Marcenko-Pastur distribution of the eigenvalues of the empirical covariance matrix (Section IV).

Our previous work [10] complied with this approach. Yet, it dealt with fast search with the Euclidean distance, and the exposed solution was secure in the 'honest but curious' model but not against Known Plaintext Attack. This weakness no longer exists in our new scheme provided the database is not too big (see Sect. VI-D).

D. The framework

The framework involves three actors: The Owner, The Server and the Client. The Owner has a set of feature vectors $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$ and a set of metadata $\mathcal{M} = \{m_n\}_{n=1}^N$. The metadata m_n associated to $\mathbf{x}_n$ can be its index n, the value of $\mathbf{x}_n$, some label (in classification application), or the ID of the content/individual from which $\mathbf{x}_n$ has been extracted, *etc*. The Owner subcontracts the search to the Server. For this purpose, he prepares offline a database $\mathcal{D}$. For privacy and security issues, this database shall not contain the vectors in the clear. The Client has a query vector $\mathbf{q}$ and he is interested in the metadata of the most similar vectors in $\mathcal{X}$. The Client is trusted either because the application only requires one-way privacy as in [7], either by assumption like in [11].

II. Fast and approximate similarity search

Our approach applies to a family of approximate similarity searches, not to only one specific technique. We describe this family in rather abstract terms now, yet sufficiently detailed to understand how it can be made more secure and private in the sequel. This description covers for instance Locally Sensitive Hashing [8], [4], [5], and variants [6], [3], as well as product quantization [9]. Security is not considered in this section.

A. A family of approximate similarity search

The indexing strategies that belong to this family all start with assuming description features lying in $\mathbb{R}^d$ and a similarity metric $\mathsf{sim}(\mathbf{q}, \mathbf{x})$ used to compare these features. This is typically a decreasing function of the Euclidean distance or a normalized correlation, $\mathbf{q}$ is the query and $\mathbf{x}$ one of the (millions to billions) database vectors.

Comparing these features in the original data space is costly, as distances are calculated over a gigantic number of features. The approximate search strategies we describe here trade response time and memory footprint for accuracy. These techniques turn original vectors into very compact signatures and turn their original similarity criterion into a fast approximation computed over their compact representations. Of course, care is taken to preserve the quality of the approximated searches compared to searching the original data.

The general principles this family of techniques adheres to are the following. The original features are first embedded into another space of a different dimensionality $\mathbb{R}^m$. The embedding is done by projecting the vectors against a random $d \times m$ matrix $\mathbf{\Pi}$: $\boldsymbol{\pi} = \mathbf{\Pi}^\top \mathbf{x}$. The resulting projections are then split into L subvectors, each of size P (for simplicity we assume $m = P \times L$, L and P being integers): $\boldsymbol{\pi} = (\boldsymbol{\pi}_1^\top, \ldots, \boldsymbol{\pi}_L^\top)^\top$.

Each subvector $\boldsymbol{\pi}_\ell$ is then quantized into Q symbols, giving $\mathbf{s}_{\mathbf{\Pi},\mathbf{x}}(\ell)$ the ℓ-th symbol of the signature. The compact signature lies therefore in the space $\{1, \ldots, Q\}^L$, compactly encoded using $L \times \lceil \log_2 Q \rceil$ bits. The codebook for quantizing each subvector might depend on ℓ.

Two similarity metrics are used to compare the signatures in the database to the query. The *symmetric approximated similarity* compares two signatures as follows:

$$\mathsf{sim}_s(\mathbf{q}, \mathbf{x}) = \sum_{\ell=1}^{L} \mathbf{T}_{s,\ell}(\mathbf{s}_{\mathbf{\Pi},\mathbf{q}}(\ell), \mathbf{s}_{\mathbf{\Pi},\mathbf{x}}(\ell)), \tag{1}$$

where $\mathbf{T}_{s,\ell}$ is a $Q \times Q$ matrix storing the typical similarity amounts between vectors falling in all pairs of quantization cells. This is of course a rather rough estimate of the true similarities, but tabulation makes it extremely fast.

The other possible metric is in contrast *asymmetric* [6], [9] as the similarity is established by comparing the signatures in the database and the projection $\boldsymbol{\pi}_q = \mathbf{\Pi}^\top \mathbf{q}$ of the query, split into L subvectors $(\boldsymbol{\pi}_{q,1}^\top, \ldots, \boldsymbol{\pi}_{q,L}^\top)^\top$. The *asymmetric approximated similarity* is then

$$\mathsf{sim}_a(\mathbf{q}, \mathbf{x}) = \sum_{\ell=1}^{L} f_\ell(\boldsymbol{\pi}_{q,\ell}, \mathbf{s}_{\mathbf{\Pi},\mathbf{x}}(\ell)). \tag{2}$$

It is efficiently computed for a large number of signatures thanks to $L \times Q$ matrix $\mathbf{T}_a$: $\mathbf{T}_a(\ell, u) = f_\ell(\boldsymbol{\pi}_{q,\ell}, u)$, so that $\mathsf{sim}_a(\mathbf{q}, \mathbf{x}) = \sum_{\ell=1}^{L} \mathbf{T}_a(\ell, \mathbf{s}_{\mathbf{\Pi},\mathbf{x}}(\ell))$.

B. A particular approximated similarity search

We instantiate this general description into a particular scheme suitable for the cosine similarity:

$$\mathsf{sim}(\mathbf{q}, \mathbf{x}) = \mathbf{q}^\top \mathbf{x} / (\|\mathbf{q}\| \|\mathbf{x}\|). \tag{3}$$

We project the normalized version of $\mathbf{x}$: $\boldsymbol{\pi} = \mathbf{\Pi}^\top \mathbf{x} / \|\mathbf{x}\|$. Then, each subvector $\boldsymbol{\pi}_\ell$ is quantized onto a set of Q predefined directions of the space $\mathbb{R}^P$. They are represented by P dimensional vectors $\{\mathbf{v}_u\}_{u=1}^Q$ s.t. $\|\mathbf{v}_u\| = cst$, $\forall u$:

$$\mathbf{s}_{\mathbf{\Pi},\mathbf{x}}(\ell) = \arg \max_{1 \le u \le Q} \boldsymbol{\pi}_\ell^\top \mathbf{v}_u. \tag{4}$$

We aim at constructing the vectors such that they are uniformly distributed over the unit sphere of $\mathbb{R}^P$. Section VI-A details how we almost achieve this for the special case $P = 8$.

Geometrically, the projection vector $\boldsymbol{\pi}_\ell$ lies inside the single hypercone of axis $\mathbf{v}_{\mathbf{s}_{\boldsymbol{\Pi},\mathbf{x}}(\ell)}$ and angle θ. Indeed, this is not true: the 'quantization cell' associated to $\mathbf{v}_u$ contains the hypercone of axis $\mathbf{v}_u$ and angle $\theta = \arccos(\max_{u' \neq u} \mathbf{v}_u^\top \mathbf{v}_{u'})/2$.

We advice the following tables for computing the metrics:

$$\mathbf{T}_{s,\ell}(u_1, u_2) = \mathbf{v}_{u_1}^\top . \mathbf{v}_{u_2}, \quad \forall (u_1, u_2) \in \{1, \ldots, Q\}^2 \quad (5)$$

$$\mathbf{T}_a(\ell, u) = \boldsymbol{\pi}_{q,\ell}^\top . \mathbf{v}_j, \ \forall (\ell, u) \in \{1, \ldots, L\} \times \{1, \ldots, Q\}. \quad (6)$$

The protocol of Section III is based on these approximated metrics with some mild changes for security reason.

III. Description of the protocol

We present a protocol based on the signature and the approximated similarities presented in Section. II-B.

A. Main structure

The main idea underlying the protocol is simple. The Client sends the signature of the query to the Server, which makes a fast but rough approximated similarity search. This results in a shortlist of candidates sent back to the Client, who makes a second and more refined similarity search. In scalable search literature, the Server usually performs this double step search. Here, the Client processes the second step as it involves confidential information about the query.

The first step is scalable (able to cope with billions of entries) but provides a crude search. What matters here is the speed for computing the approximated similarities, and the size R_S of the shortlist so that the most similar vectors are almost surely in, even if they are not ranked first.

In the second step, the computation of the similarities is slower, which is less important since only $R_S \ll N$ candidates remain. What matters here is the recall at rank $R \leq R_S$, denoted by 'l-recall@R'. This is the average ratio of the number of true l-most similar vectors among the R first returned vectors over l. As usual in approximate search literature, we focus on the 1-recall@R, which is the probability that the most similar entry is in the first R returned vectors.

B. Offline preparation of the Owner

The protocol starts as follows. Offline, the Owner processes the vectors of $\mathcal{X}$ together with the set of metadata $\mathcal{M}$ to prepare the database $\mathcal{D}$ to be given to the Server. We constrain the construction of the set $\{\mathbf{v}_u\}_{u=1}^Q$ to contain antipodal vectors: Q is even and the vectors can be grouped into $Q/2$ pairs such that $\mathbf{v}_{2j} = -\mathbf{v}_{2j-1}$, $\forall j \leq Q/2$. The Owner first draws a set of K secret matrices $\{\boldsymbol{\Pi}^{(k)}\}_{k=1}^K$ and also a secret key sk for a symmetric cryptosystem, like AES.

The signature of $\mathbf{x}_n$ with $\boldsymbol{\Pi}^{(k)}$ is slightly different than (4):

$$\mathbf{s}'_{\boldsymbol{\Pi}_k,\mathbf{x}_n}(\ell) = \left\lfloor \frac{\mathbf{s}_{\boldsymbol{\Pi}_k,\mathbf{x}_n}(\ell) + 1}{2} \right\rfloor \quad \forall \ell, \, 1 \leq \ell \leq L. \quad (7)$$

This symbol pertains to $\{1, \ldots, Q/2\}$. The quantization cell is the two nappes hypercone of axis $\mathbf{v}_{2*\mathbf{s}'_{\boldsymbol{\Pi}_k,\mathbf{x}_n}(\ell)}$ (or equivalently $\mathbf{v}_{2*\mathbf{s}'_{\boldsymbol{\Pi}_k,\mathbf{x}_n}(\ell)-1}$). This signature is encoded with $L \times \lceil \log_2(Q/2) \rceil$ bits. In addition, the Owner computes the following side information:

$$\mathbf{s}''_{\boldsymbol{\Pi}_k,\mathbf{x}_n}(\ell) = \mod (\mathbf{s}_{\boldsymbol{\Pi}_k,\mathbf{x}_n}(\ell), 2) \quad \forall \ell, \, 1 \leq \ell \leq L. \quad (8)$$

This bit indicates in which single nappe hypercone the projection belongs to: $\boldsymbol{\pi}_{\mathbf{x}_n,\ell}^\top . \mathbf{v}_{2*\mathbf{s}'_{\boldsymbol{\Pi}_k,\mathbf{x}_n}(\ell) - \mathbf{s}''_{\boldsymbol{\Pi}_k,\mathbf{x}_n}(\ell)} \geq 0$. Finally, the Owner appends the following information into an entry of database $\mathcal{D}$: $[\mathbf{s}'_{\boldsymbol{\Pi}_k,\mathbf{x}_n}, \mathsf{enc}_{sk}([k, \mathbf{s}''_{\boldsymbol{\Pi}_k,\mathbf{x}_n}, m_n])]$, where $\mathsf{enc}_{sk}(\cdot)$ is the encryption with secret key sk and $[a, b]$ is the concatenation of strings a and b. In other words, the Server has access to the compact representation $\mathbf{s}'_{\boldsymbol{\Pi}_k,\mathbf{x}_n}$, but not the side-information pieces k, $\mathbf{s}''_{\boldsymbol{\Pi}_k,\mathbf{x}_n}$ and m_n.

These $K.N$ entries are shuffled before sending $\mathcal{D}$ to the Server such that this latter has no clue on the indices n and k that produced a given entry. The Owner also sends the Server the $Q/2 \times Q/2$ matrix $\mathbf{T}_s$ necessary to compute the symmetric approximated similarity (1):

$$\mathbf{T}_s(u_1, u_2) = |\mathbf{v}_{2*u_1}^\top . \mathbf{v}_{2*u_2}|. \quad (9)$$

Note that the vectors of $\mathcal{X}$, the secret matrices $\{\boldsymbol{\Pi}_k\}_{k=1}^K$, and the secret key sk are not disclosed to the Server.

C. Approximated search at the Server side

Online, the Client follows the same process. For one query vector $\mathbf{q}$, a 'bag' of K signatures $\{\mathbf{s}'_{\boldsymbol{\Pi}_k,\mathbf{q}}\}_{k=1}^K$ are computed and sent to the Server. However, there is no specific order in the bag, so that the Server does not know which secret key has generated which signature inside the bag.

The Server proceeds the symmetric approximated search for all signatures in the query bag over all the signatures in the dataset. This has a complexity of $O(K^2.N.L)$. The Server gives back K shortlists of size R_S, one per signature in the query bag respecting the order found in the bag. Each element of a shortlist is a full entry of $\mathcal{D}$, i.e. $[\mathbf{s}'_{\boldsymbol{\Pi}_k,\mathbf{x}_n}, \mathsf{enc}_{sk}([k, \mathbf{s}''_{\boldsymbol{\Pi}_k,\mathbf{x}_n}, m_n])]$.

D. Approximated search at the Client side

The Client receives a shortlist per signature in the query bag. The Client knows which secret matrices $\boldsymbol{\Pi}_k$ generated the query signature giving birth to a given shortlist. For a given k, he decrypts the side-information in the corresponding shortlist and prunes out any entry based on another secret key: This is a false positive, as it is a match between two signatures computed with different secret matrices. For the remaining vectors, the Client computes the asymmetric approximated similarity (2) via (6). This is possible thanks to the side-information $\mathbf{s}''_{\boldsymbol{\Pi}_k,\mathbf{x}_n}$. When all the shortlists have been processed, the Client sorts in decreasing order all the approximated similarities to obtain the final ranking.

IV. A primer on statistical signal processing over large random matrices

We introduce some theoretical elements on statistics of large random matrices. This section is somehow disconnected from the previous ones, but it will be the foundation of the security assessments of our protocol in Sect. V.

A. The 'Information plus Noise' model

Suppose we observe N_o signals in $\mathbb{R}^d$ in the form $\mathbf{x}_n = \mathbf{A}.\mathbf{s}_n + \mathbf{n}_n$, where $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_K)$ is a fixed but unknown with a limited rank $K \ll d$ matrix whose columns have unit norm, $\mathbf{s}_n \in \mathbb{R}^K$ are random source signals mutually independent with power $\rho > 0$, and $\mathbf{n}_n \in \mathbb{R}^d$ is a white noise of unit variance, not necessary Gaussian distributed. This setup is often called the 'Information plus Noise' model. Let stack the observations in one matrix, $\mathbf{X}_{N_o} = (\mathbf{x}_1, \ldots, \mathbf{x}_{N_o})$, and as for the noise $\mathbf{N}_{N_o} = (\mathbf{n}_1, \ldots, \mathbf{n}_{N_o})$. A well known estimator of $\mathbf{A}$ is to proceed a PCA, which amounts to make the SVD of the empirical covariance matrix $\hat{\mathbf{R}}_{N_o} = \mathbf{X}_{N_o}.\mathbf{X}_{N_o}^\top/N_o$. This works because, for a fixed d, $\hat{\mathbf{R}}_{N_o}$ converges as $N_o \to \infty$ to the true covariance matrix $\rho\mathbf{A}\mathbf{A}^\top + \mathbf{I}_d$. $(d-K)$ eigenvalues equals 1, and K others $1+\rho$, so that the eigenvectors associated to the biggest eigenvalues reveal the K-dimensional subspace spanned by the columns of $\mathbf{A}$.

In practice, this may totally fail if N_o is not much bigger than d. This has been studied with the following theoretical model: K is fixed while $d = c.N_o$ ($c < 1$) with $N_o \to \infty$. Let us first focus on the 'Noise' component. A surprising result is that its empirical covariance matrix $\mathbf{N}_{N_o}\mathbf{N}_{N_o}^\top/N_o$ no longer converges towards $\mathbf{I}_d$ as $N_o \to \infty$. Indeed, the eigenvalues of $\mathbf{N}_{N_o}\mathbf{N}_{N_o}^\top/N_o$ are random variables whose law is known as the Marcenko-Pastur distribution defined over the interval $[(1-\sqrt{c})^2, (1+\sqrt{c})^2]$.

B. Indetectability of the eigenvalues

This may have a huge impact on PCA based detector or estimator. A first result from [2] states that if $\rho > \sqrt{c}$, it is possible to estimate the eigenvalues of $\mathbf{A}\mathbf{A}^\top$ from the K biggest eigenvalues of $\hat{\mathbf{R}}_{N_o}$. But, if this condition is not met, the eigenvalues related to the 'Information' part remain hidden among the eigenvalues due the 'Noise' part. For instance, [2] showed that even the Generalized Likelihood Ratio Test cannot make the distinction between 'Noise' only (i.e. $\rho = 0$) and 'Information plus Noise' ($0 < \rho < \sqrt{c}$) observations.

C. Unreliable estimation by the eigenvectors

Even if $\rho > \sqrt{c}$, a further difficulty raises in the estimation of the subspace spanned by the columns of $\mathbf{A}$. Even for the simple case where $K = 1$ (i.e. $\mathbf{A} = \mathbf{a}_1$), [12] shows that the eigenvector $\hat{\mathbf{a}}_1$ associated to the largest eigenvalue of $\hat{\mathbf{R}}_{N_o}$ is a bad estimator because $\kappa = \mathbf{a}_1^\top\hat{\mathbf{a}}_1 \to \sqrt{\frac{1-c/\rho}{1+c/\rho}} < 1$ as $N_o \to \infty$. In other words, since its norm is one, the true vector $\mathbf{a}_1$ lies in the intersection of the unit sphere and the hyperplan $\mathbf{x}^\top\hat{\mathbf{a}}_1 = \kappa$. In a high dimensional space, this makes a big ambiguity whenever κ cannot go to 1.

V. Security assessments

When subcontracting the search to the Server, the Owner may have two worries: i) the Server might guess the most similar vector to a given entry of the database or to the query signature, ii) the Server can reconstruct either the query or the vectors from their signatures.

A. Guessing the most similar vector

The first threat pertains to the honest but curious model. Knowing $\mathcal{D}$, the Server can infer which vectors are more similar than others. This threat is absolutely impossible in the past approaches based on heavy cryptographic primitives because these encryption schemes are semantically secure. Yet, in our system, this threat is mitigated since the metadata (the value of importance) associated to each vector are encrypted. The Server doesn't know sk as the 'honest but curious' model excludes any collusion between Client and Server.

Yet, the Server can still guess the most similar vector based on the signatures $\mathbf{s}'_{\mathbf{\Pi}_k,\mathbf{x}_n}$. Starting from one entry of $\mathcal{D}$ or the query signature, the symmetric approximated similarity of (1) yields a list of most similar vectors. There is no need of the secret matrices, the knowledge of $\mathbf{T}_s$ defined in (9) is enough. Yet, since the signatures are computed with different secret matrices, plenty of false positives indeed spoil the quality of the search. The feasibility of this attack is gauged by the 1-recall@R obtained with the symmetric approximated similarity. Fig. 1 shows that this performs much worse than the similarity search at the Client side.

B. Reconstruction of the vectors

The second threat is possible if the projection matrix is disclosed via the following equation:

$$\hat{\mathbf{x}} = \mathbf{\Pi}_k^\dagger \left(\mathbf{v}^\top_{\mathbf{s}_{\mathbf{\Pi}_k,\mathbf{x}}(1)}, \ldots, \mathbf{v}^\top_{\mathbf{s}_{\mathbf{\Pi}_k,\mathbf{x}}(L)}\right)^\top, \tag{10}$$

where $\mathbf{\Pi}^\dagger$ is the Moore-Penrose pseudo-inverse of $\mathbf{\Pi}$. This reconstruction is lossy due to the quantization process.

This threat is impossible in the honest but curious model since the Server does not know any of the secret matrices. However, the literature often extends this model and argues that the usual homomorphic encryption schemes are secure under a Known Plaintext Attack. In our context, this translates into the following requirement: even if the Server could observe N tuples $\{(\mathbf{x}_n, \{\mathbf{s}'_{\mathbf{\Pi}_k,\mathbf{x}_n}\}_k)\}_{n=1}^N$, he should not be able to disclose any secret matrix $\mathbf{\Pi}_k$. Note that the Server observes vectors, not their quantized version.

The idea is the following. The curious Server chooses two integers $1 \le u \le Q/2$ and $1 \le \ell \le L$ and makes the following group: $\mathcal{X}_{u,\ell} = \{\mathbf{x}_n | \exists k, \mathbf{s}'_{\mathbf{\Pi}_k,\mathbf{x}_n}(\ell) = u\}$. These vectors in $\mathbb{R}^d$ share the property of being strongly oriented along the direction $\pm\mathbf{\Pi}_1^{(\ell)}\mathbf{v}_{2u}$, and/or $\pm\mathbf{\Pi}_2^{(\ell)}\mathbf{v}_{2u}$, ..., and/or $\pm\mathbf{\Pi}_K^{(\ell)}\mathbf{v}_{2u}$. This means that more power lies in the K-dimensional subspace $\mathcal{S}_{u,\ell}$ spanned by $(\mathbf{\Pi}_1^{(\ell)}\mathbf{v}_u, \ldots, \mathbf{\Pi}_K^{(\ell)}\mathbf{v}_u)$, and this should be noticeable in their covariance matrix.

This is where the theoretical elements of Section IV come into the picture. In expectation, the number of useful observations over all $\mathcal{X}$, i.e. the size of set $\mathcal{X}_{u,\ell}$, equals $N_o(K) = N(1-(1-P_1)^K)$, where P_1 is the probability that $\mathbf{s}'_{\mathbf{\Pi}_k,\mathbf{x}_n}(\ell) = u$ for a given k. Section IV tells that a first factor of utmost importance is the ratio $c = d/N_o(K)$. The second factor of importance is the power ρ_K of the equivalent 'Information' part. The annex shows that as K increases, this power vanishes (see (16)). It remains to fine-tune a setup, and especially the

parameter K, enforcing $\rho_K < \sqrt{d/N_o(K)}$. This assesses that the disclosure of the subspace $\mathcal{S}_{u,\ell}$ is impossible, and consequently the secret matrices $\{\mathbf{\Pi}_k\}_{k=1}^K$.

The initial assumption was that the Server knows no more than N vectors and their bag of signatures. This holds if the secret matrices encode only one set $\mathcal{X}$ of size N. The security of our protocol is deeply related to the size of the database.

VI. Experimental proof of concept

A. Implementation details

We present which $\{\mathbf{v}_u\}_{u=1}^Q$ we choose and how we quantize the direction of subvectors. To do so efficiently, we set $P = 8$, and we quantize $\boldsymbol{\pi}_\ell/\|\boldsymbol{\pi}_\ell\|$ to the nearest point of the E_8 lattice. It appears that any point on the sphere of radius 1 is quantized onto one of the 240 points of the first shell of the lattice E_8. These are the lattice points whose norm equal $\sqrt{2}$. By doing so, we take advantage of the fast quantization algorithm onto lattice E_8. These vectors make angles with each other of values $\{0, \pi/3, \pi/2, 2\pi/3, \pi\}$. Any subvector making an angle smaller that $\pi/6$ with $\mathbf{v}_u$ is then quantized to this lattice point.

B. Runtime and communication payload

The setup is as follows: $\mathcal{X}$ is composed of $N = 50,000$ white Gaussian vectors of dimension $d = 256$. There are $N_q = 400$ queries. The size of the shortlists is $R_S = 200$. The default parameters are set to $(L, P, K) = (512, 8, 8)$ (unless explicitly stated). Runtimes are given for a Matlab implementation running on a 2.4 GHz Intel Core i7 platform (single thread). The Server returns the shortlists within 2.0 sec. The Client re-ranks the relevant vectors within 0.07 sec. As for the payload of the communication, querying costs 3.6 KB (the size of a signature) whereas returning the shortlists amounts to 820 KB. These orders of magnitude are in between those reported with similar environment in the literature of

- secure computation: for a signature of 1.5 KB, the secure computation of a single similarity (1 vs. 1) takes 60 s and 393 MB of communication [11, Tab. III].
- fast similarity search: for a signature of 8B, the search 1 vs. 10^6 takes 40 ms (1-recall@100=0.65) [9, Tab. V].

C. Client vs. curious Server performances

Fig. 1 shows the huge gap between the quality of the approximate search at the Client and (curious) Server sides. The 1-recall@R of the Client is limited by the probability that the most similar vector pertains to the shortlists. This limit is reached around $R = 20$ proving the re-ranking is very powerful thanks to the asymmetric approximated search. At the Server side, the median of the rank of the most similar vector is 600 for long signatures ($L = 512$), and bigger than 10^3 for short signatures ($L = 384$). In other words, the Server has almost no clue of which vector the Client is looking for, and this enables the one-way privacy of the search.

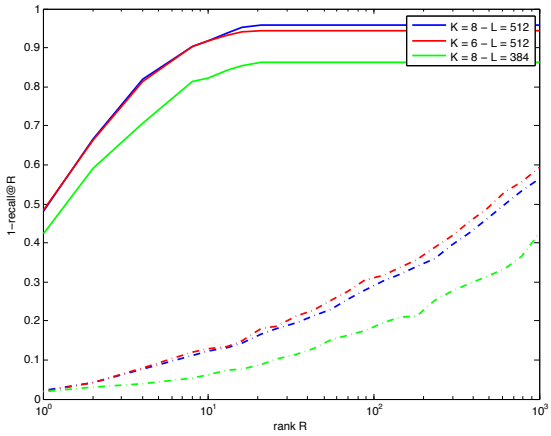

Fig. 1. Client (plain) and the curious Server's (dashed) 1-recall@R.

TABLE I
Power of the 'Information' Part and Maximum Number of Vectors in the Database per Dimension. $P = 8$, $\theta = \pi/6$.

K	6	7	8	9	10
ρ_K	0.91	0.78	0.68	0.61	0.55
$N_{\lim}(K)/d$	159	186	212	239	265

D. Security assessment

The numerical application of (14) (see Appendix I-A) with $P = 8$, $\theta = \pi/6$ and $K = 1$ gives a power of 'Information' part of $\rho_1 \approx 5.43$, which is not lower than $c < 1$. We need at least $K = 6$ secret matrices (see Table I) to dilute this power s.t. $\rho_K < 1$ (see (16)). Then, the 'Information' part can be hidden into the 'Noise' if $\rho_K < \sqrt{d/N_o(K)}$. This in turn proves security if these secret matrices are used for only one database whose size is smaller than $N_{\lim}(K)$:

$$N_{\lim}(K) = \frac{d}{\rho_K^2(1-(1-P_1)^K)}. \tag{11}$$

Fig. 2 illustrates this statistical phenomenon with one database size lower than $N_{\lim}(K)$ and the other infringing this limit. In the latter case, the Marcenko-Pastur interval is smaller and the eigenvalues of the 'Noise' do not hide those related to the 'Information'. The distribution of eigenvalues of the empirical covariance matrix significantly differs from the Marcenko-Pastur distribution. Table I shows that $N_{\lim}(K)$ increases with K. Yet, this is limited by the fact that Sec. IV assumed $K \ll d$.

VII. Conclusion

We propose a protocol for similarity search in high-dimensional space striking a trade-off between speed and security. The assumption is that the Client is trusted and the Owner outsources the search to a honest but curious Server. The security against Known Plaintext Attack and one-way privacy assessments are based on statistical considerations. The protocol is much faster and consumes far less bandwidth than solutions based on homomorphic cryptography. However, security has a prize to pay: It restricts the size of the database

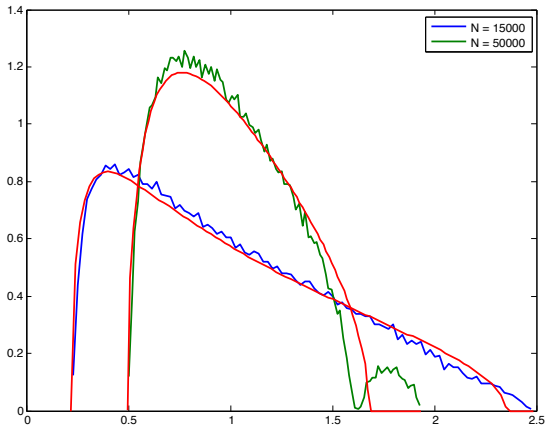

Fig. 2. Empirical density distribution of the eigenvalues of the covariance matrix of $\mathbf{x} \in \mathcal{X}_{q,\ell}$ vs. the Marcenko-Pastur p.d.f. (red). In green, the eigenvalues due to the 'Information' are isolated because $N > N_{\lim}(K)$.

and precludes the scalability to be as large as current state-of-the-art in fast (but unsecure) similarity search. The protocol cannot resist a Chosen Plaintext Attack.

APPENDIX I

A. Power distribution with one hypercone

Let us study a mathematical model $\mathbf{x} = \mathbf{\Pi}\boldsymbol{\pi} + \mathbf{\Pi}^{\perp}\boldsymbol{\pi}^{\perp}$, where $(\mathbf{\Pi}, \mathbf{\Pi}^{\perp})$ forms a basis of $\mathbb{R}^d$, whose P first vectors are gathered in $\mathbf{\Pi}$. We suppose that $\mathbf{x}$ is distributed as a Gaussian white noirse, therefore so are $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^{\perp}$ in their subspace.

The probability P_1 that $\boldsymbol{\pi}$ lies in the two nappes hypercone $\mathcal{C}(\mathbf{v}, \theta)$ of axis $\mathbf{v}$ and angle θ equals:

$$P_1 = \mathbb{P}(\boldsymbol{\pi} \in \mathcal{C}(\mathbf{v}, \theta)) = 1 - I_{\cos^2(\theta)}(1/2, (P-1)/2)), \quad (12)$$

where $I(\cdot)$ is the regularized incomplete beta function.

We now compute the covariance matrix of $\mathbf{x}$ knowing that $\boldsymbol{\pi} \in \mathcal{C}(\mathbf{v}, \theta) \subset \mathbb{R}^P$. This is modeled by $\boldsymbol{\pi} = a.\mathbf{v}/\|\mathbf{v}\| + \mathbf{n}$ with $\mathbf{v}^{\top}\mathbf{n} = 0$ and $a^2 > \|\mathbf{n}\|^2/\tan^2(\theta)$. While a is distributed as $\mathcal{N}(0,1)$ (density $f(\cdot)$), $\|\mathbf{n}\|^2$ is χ^2_{P-1} distributed (density $g(\cdot)$). In $\mathbb{R}^P$, denote the power along the direction $\mathbf{v}$ by $\bar{\lambda}_1$. We have $\bar{\lambda}_1 = \mathbb{E}(a^2)$ with:

$$\mathbb{E}(a^2) = \left(\int_0^{+\infty} \int_{\sqrt{n}/\tan(\theta)}^{+\infty} a^2 f(a) g(n)\, da dn \right) / P_1. \quad (13)$$

An integration by parts gives $\bar{\lambda}_1 = 1 + \rho_1$ with

$$\rho_1 = \frac{\tan^{-1}(\theta)}{(1+\tan^{-2}(\theta))^{\frac{P}{2}}} \frac{\Gamma\left(\frac{P}{2}\right)}{\sqrt{\pi}\Gamma\left(\frac{P-1}{2}\right)} P_1^{-1}. \quad (14)$$

Denote the power along a direction orthogonal to $\mathbf{v}$ by $\underline{\lambda}_1$:

$$\underline{\lambda}_1 = 1 - \frac{P_1}{1-P_1}\bar{\lambda}_1 = 1 - \frac{P_1}{1-P_1}(1+\rho_1). \quad (15)$$

Mapping back to $\mathbb{R}^d$, vector $\mathbf{x}$ whose projection $\boldsymbol{\pi} \in \mathcal{C}(\mathbf{v}, \theta)$ has a covariance matrix with one eigenvalue equaling $\bar{\lambda}_1$, $(P-1)$ eigenvalues $\underline{\lambda}_1$, and $(d-P)$ unit eigenvalues set to 1.

B. Power distribution with K hypercones

Suppose that $\mathbf{x} \in \mathbb{R}^d$ is such that there exists at least one matrix $\mathbf{\Pi}_k$ (over a set of K) giving $\mathbf{\Pi}_k^{\top}\mathbf{x} \in \mathcal{C}(\mathbf{v}, \theta)$. This happens with a probability $P_K = 1 - (1-P_1)^K \approx KP_1$ for small P_1. We suppose that $\mathbf{\Pi}_{k_1}^{\top}.\mathbf{\Pi}_{k_2} = \delta_{k_1}(k_2)\mathbf{I}_P$, which is feasible if $KP \leq d$. $(\mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_K, \mathbf{\Pi}'^{\perp})$ is then a base of $\mathbb{R}^d$. For a fixed k, such a vector $\mathbf{x}$ has a projection $\mathbf{\Pi}_k^{\top}\mathbf{x} \in \mathcal{C}(\mathbf{v}, \theta)$ with probability P_1/P_K, and its power along direction $\mathbf{\Pi}_k\mathbf{v}$ equals: $\bar{\lambda}_K = \bar{\lambda}_1 P_1 P_K^{-1} + (1 - P_1 P_K^{-1}) = 1 + \rho_K$, with

$$\rho_K = \rho \frac{P_1}{P_K} \approx \frac{\rho}{K}. \quad (16)$$

By the same token, for any vector $\mathbf{v}^{\perp} \in \mathbb{R}^P$ s.t. $\mathbf{v}^{\top}\mathbf{v}^{\perp} = 0$, the power along the direction $\mathbf{\Pi}_k\mathbf{v}^{\perp}$ follows:

$$\underline{\lambda}_K = 1 - \frac{P_1^2(1+\rho)}{P_K(1-P_1)} \approx 1 - K^{-1}\frac{P_1}{1-P_1}(1+\rho). \quad (17)$$

In the end, the covariance matrix of such vector $\mathbf{x}$ has K eigenvalues equalling $\bar{\lambda}_K$, $(P-1)K$ eigenvalues $\underline{\lambda}_K$, and $(d-KP)$ unit eigenvalues. Eq. (16) and (17) show that as K increases, $(\underline{\lambda}_K, \bar{\lambda}_K)$ are closer to 1. Therefore, they become hidden into the Marcenko-Pastur interval when the number of observations is not big compared to d.

REFERENCES

[1] M. Barni, T. Kalker, and S. Katzenbeisser. Inspiring new research in the field of signal processing in the encrypted domain. *IEEE Signal Processing Magazine*, 30(2), March 2013.

[2] P. Bianchi, M. Debbah, M. Maida, and J. Najim. Performance of statistical tests for single-source detection using random matrix theory. *Information Theory, IEEE Transactions on*, 57(4):2400–2419, 2011.

[3] P. Boufounos and S. Rane. Secure binary embeddings for privacy preserving nearest neighbors. In *IEEE Int. Workshop on Information Forensics and Security (WIFS)*, pages 1 –6, dec. 2011.

[4] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proc. of the 34th ACM symposium on Theory of computing*, STOC '02, pages 380–388, New York, NY, USA, 2002. ACM.

[5] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-Sensitive Hashing scheme based on p-stable distributions. In *Proc. Symposium on Computational Geometry*, pages 253–262, 2004.

[6] W. Dong, M. Charikar, and K. Li. Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces. In *Proc. of the 31st ACM SIGIR conference on Research and development in information retrieval*, pages 123–130, New York, NY, USA, 2008. ACM.

[7] G. Fanti, M. Finiasz, and K. Ramchandran. One-way private media search on public databases: The role of signal processing. *Signal Processing Magazine, IEEE*, 30(2):53–61, 2013.

[8] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM.

[9] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 33(1):117 –128, jan. 2011.

[10] B. Mathon, T. Furon, L. Amsaleg, and J. Bringer. Secure and Efficient Approximate Nearest Neighbors Search. In ACM, editor, *Proceedings of the first ACM workshop on Information hiding and multimedia security*, IH & MMSec '13, pages 175–180, Montpellier, France, June 2013.

[11] J. Troncoso-Pastoriza, D. Gonzalez-Jimenez, and F. Perez-Gonzalez. Fully private noninteractive face verification. *Information Forensics and Security, IEEE Transactions on*, 8(7):1101–1114, 2013.

[12] P. Vallet, P. Loubaton, and X. Mestre. Improved subspace estimation for multivariate observations of high dimension: The deterministic signals case. *Information Theory, IEEE Transactions on*, 58(2):1043–1068, 2012.