DOI 10.1007/s10559-016-9899-x
Cybernetics and Systems Analysis, Vol. 52, No. 6, November, 2016

NEW TOOLS OF CYBERNETICS, INFORMATICS, COMPUTER ENGINEERING, AND SYSTEMS ANALYSIS

REAL-VALUED EMBEDDINGS AND SKETCHES FOR FAST DISTANCE AND SIMILARITY ESTIMATION

D. A. Rachkovskij

UDC 004.22+004.93′11

Abstract. *This survey article considers methods and algorithms for fast estimation of data distance/similarity measures from formed real-valued vectors of small dimension. The methods do not use learning and mainly use random projection and sampling. Initial data are mainly high-dimensional vectors with different measures of distance (Euclidean, Manhattan, statistical, etc.) and similarity (dot product, etc.). Vector representations of non-vector data are also considered. The resultant vectors can also be used in similarity search algorithms, machine learning, etc.*

Keywords: *distance, similarity, embedding, sketch, dimensionality reduction, random projection, sampling, Johnson–Lindenstrauss lemma, kernel similarity, similarity search.*

1. BASIC CONCEPTS

Distance and similarity functions (measures) are widely used in similarity search and in many applications of data analysis, machine learning, and statistics (cluster analysis, classification and approximation by nearest-neighbor methods, multidimensional scaling, etc.). For complicatedly computable distances and similarities, their fast estimation or computation of bounds on their values is topical. To obtain such an estimate, initial (vector and non-vector) representations of data (objects) of different types with different distance/similarity measures are often transformed into representations (usually vector representations of small dimension) that make it possible to easily compute estimates for the similarity of initial data. The complexity of estimation of distances/similarities between vectors (for example, Euclidean distance, dot product, etc.) is linearly dependent on vector dimension and, hence, complexity is small in the case of small dimensions. For vector representations, there also are many methods for similarity search, statistical pattern recognition, classification, clusterization, approximation, feature selection, etc.

This article presents a survey of approaches, methods, and algorithms for fast estimation of distances/similarities between initial data representations from real-valued vector representations. (Binary and integer-valued vector representations for fast distance/similarity estimation are considered in [1].) Methods without adaptation to data are mainly presented (but see Sec. 9.4). The majority of the considered methods and algorithms are practically implementable, though only theoretical bounds on parameter values are given in certain cases. (As a result of a limitation on the size of this article, publications containing references to previous papers are mainly cited.)

1.1. Distances and similarities. For each type of representation of data (objects), there are different distance/similarity measures. The number of types of representations is small. Vector representations and also sets, sequences, trees, and graphs are most widespread.

International Scientific-Educational Center of Information Technologies and Systems, NAS and MON of Ukraine, Kyiv, Ukraine, *dar@infrm.kiev.ua*. Translated from Kibernetika i Sistemnyi Analiz, No. 6, November–December, 2016, pp. 156–180. Original article submitted May 13, 2016.

For example, if objects are represented in the form of a collection of numerical features (as real-valued vectors **x** and **y** of dimension D), then the dot product $\mathrm{sim}_{\mathrm{dot}}(\mathbf{x}, \mathbf{y}) \equiv \langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{D} x_i y_i$ can be used to estimate similarity. Larger values of similarities correspond to more similar objects. To estimate similarities of objects, distance is also used, i.e., "dissimilarity" (for example, the angle or Euclidean distance between vectors). To large similarity values correspond small distance values. Many distances are metrics, i.e., satisfy metric axioms such as the triangle inequality, etc. Properties of similarity measures can be used to accelerate similarity search, for example, the triangle inequality is used for metrics.

For vectors, Minkowski distances L_s of different order s: $\|\mathbf{x} - \mathbf{y}\|_s = \left(\sum_{i=1}^{D} |x_i - y_i|^s \right)^{1/s}$ are widely used. The metric distances L_2 (Euclidean $\|\mathbf{x} - \mathbf{y}\|_2$), L_1 (Manhattan $\|\mathbf{x} - \mathbf{y}\|_1$), and L_∞ (the Chebyshev distance or the maximum of $\|\mathbf{x} - \mathbf{y}\|_\infty$) are most widespread. When $0 < s < 1$, fractional distances are obtained that are not metrics. The complexity of computing the Minkowski distance between two vectors amounts to $O(D)$. We also denote a vector space with a distance L_s by L_s (or L_s^D when its dimension equals D). Many types of distances/similarities for different representations of objects are considered in [2]. In this survey, definitions of distances are introduced as needed.

1.2. Embeddings. A transformation $\mathrm{f}(x)$ of the set of objects of some initial space into a target (usually a "simpler") space preserving the initial distances is called an embedding. Though spaces of various types (for example, tree metrics [3], etc.) can be target spaces, embeddings into normalized (vector) spaces L_s are mainly used.

The quality of embeddings is estimated [3, 4] by the amount of distance distortions. For small $\varepsilon > 0$, a multiplicative distortion, i.e., a minimal ε, is often used for which

$$(1-\varepsilon)\mathrm{dist}_1(x, y) \le \mathrm{dist}_2(\mathrm{f}(x), \mathrm{f}(y)) \le (1+\varepsilon)\mathrm{dist}_1(x, y). \quad (1)$$

We call it $1 \pm \varepsilon$-distortion or $1+\varepsilon$ distortion. For large distortions $A > 1$, the expression for the multiplicative A-distortion is of the form

$$\mathrm{dist}_1(x, y)/\sqrt{A} \le \mathrm{dist}_2(\mathrm{f}(x), \mathrm{f}(y)) \le \sqrt{A}\,\mathrm{dist}_1(x, y). \quad (2)$$

We define an additive distortion $\pm\varepsilon_a$, $\varepsilon_a > 0$, as

$$\mathrm{dist}_1(x, y) - \varepsilon_a \le \mathrm{dist}_2(\mathrm{f}(x), \mathrm{f}(y)) \le \mathrm{dist}_1(x, y) + \varepsilon_a. \quad (3)$$

Distortions can also be defined not only for distances dist but also for similarities sim.

When $\varepsilon = 0$, $\varepsilon_a = 0$, and $A = 1$ in expressions (1)–(3), distances/similarities are preserved exactly, i.e., isometry takes place; for example, the Frechet embedding of N objects of a metric space into L_∞, embedding of the entire space L_1 into L_∞ ([3] and Sec. 6.1), and embedding of kernel similarities $\kappa(x, y)$ into a Hilbert space H (Sec. 7).

Drawbacks of isometric embeddings are considered to be as follows: distances are usually preserved only for a given set of objects, dimensions of embeddings are large, and only a few of such embeddings are known. Therefore, approximate embeddings into vector spaces (of small dimension) are needed since they allow one to quickly estimate initial distances with small distortions [3, 4].

For problems such as similarity search, query objects are often unknown in advance, and the composition of objects of the base in which the search is performed can vary. Therefore, oblivious methods of formation of object representations that can be applied to new objects [5] (without changing existing representations) are required. In [6], oblivious embeddings of objects do not depend on other objects (a stronger definition of obliviousness). In this survey, the majority of transformations being considered are oblivious according to [6].

Since the embedding of any objects with small distortions is a difficult task, oblivious embeddings are usually randomized (are carried out using pseudo-random numbers and guarantee distortions only with some probability). For example, dimensionality reduction of vectors (an embedding version in which the form of the distance function is preserved and the dimension of representations decreases) is randomized for the Euclidean distance according to the JL lemma with the help of random projections (Sec. 2).

1.3. Sketches. Compact representations of initial objects that are used to estimate some of their characteristics are called sketches [7]. As well as embeddings, sketches are used to estimate distances/similarities and usually are vectors.

To estimate distances between initial objects from sketches, not only distance (as in embeddings) but also other characteristics can be used (for example, median estimators, etc., see Sec. 4). The analytical dependence of the initial distance/similarity on some quantity determined from sketches turns out to be complicated or unknown, and tabulation can be used for obtaining estimates.

Compact representations used in streaming processing [7] (when object representations are specified by a sequence of components or their increments) are often called sketches. Vectors with binary or integer-valued (discrete) components [1] are also called sketches. Such vectors are easily processed and usually occupy less memory than initial representations of objects. In this article, we call sketches vector representations of initial objects obtained with a view to estimating distances/similarities between them (including the results of embeddings).

The time of estimation of the initial distance/similarity from sketches of dimension d usually amounts to $O(d)$ (linear time complexity of an algorithm). Therefore, for $d< D$ initial vectors, we obtain accelerated estimation. For initial representations with a complexity lower than the linear complexity of computing distance/similarity, its reduction to linear complexity for sketches also is a source of acceleration. Note that embeddings and sketches turn out to be useful without reduction in the dimensionality (the number of elements) of representations, for example, when, for them, there are efficient algorithms of similarity search, estimation of distance/similarity measures, or other algorithms or methods requiring the use of representations of the obtained type [4].

1.4. Structure of this survey article. Section 2 considers the dimensionality reduction of vectors of a Euclidean space by random projection and distortions of estimates for the Euclidean distance (and also for the dot product of and angle) between initial vectors on the basis of obtained sketches. In Sec. 3, the acceleration of random projections is discussed. In Sec. 4, embeddings and sketches are given for estimating non-Euclidean Minkowski distances.

Section 5 considers sketches obtained by sampling (selection of a subset of components of initial representations), Section 6 describes embeddings for estimating distances between non-vector data, and Section 7 examines the approximation of kernel similarities.

In Sec. 8 other lines of investigation are described including embeddings of statistical distances, equivalence of sketches and embeddings, and approximate similarity search. In Sec. 9, advantages and drawbacks of the considered real-valued embeddings and sketches are considered and a comparison with learning-based methods is given.

2. DIMENSIONALITY REDUCTION OF VECTORS OF A EUCLIDEAN SPACE BY RANDOM PROJECTION

2.1. Johnson–Lindenstrauss lemmas. The possibility of estimation of the Euclidean distance L_2 between initial vectors with small distortion in terms of the Euclidean distance between their embeddings of small dimension (i.e., dimensionality reduction) is provided by the Johnson of–Lindenstrauss (JL) lemma.

JL LEMMA [8, 4]. For a set $\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$ of any N vectors in a real space $\mathbb{R}^D$ (irrespective of its dimension D), there is an embedding $\mathrm{f}:\mathbb{R}^D \to \mathbb{R}^d$, $d=O(\log N/\varepsilon^2)$, with a distortion of each of $N(N-1)/2$ distances L_2 between these vectors by at most $1\pm\varepsilon$ (i.e., for them, condition (1) is fulfilled for the Euclidean dist_1 and dist_2).

In proofs of the JL lemma [9–11], linear randomized embeddings are used whose properties are determined by versions of the so-called distributional JL lemma (DJL), which is also well known as the lemma of random projections [4]. These DJL lemmas assert that there are classes of distributions of matrices of size $d\times D$ such that a matrix $\mathbf{R}$ randomly chosen from a distribution, for any vector $\mathbf{z}\in\mathbb{R}^D$ with probability Pr (over instances of $\mathbf{R}$), provides the multiplicative $1\pm\varepsilon$ distortion of the Euclidean norm $\mathbf{z}$: $\Pr\{(1-\varepsilon)||\mathbf{z}||_2 \le ||\mathbf{Rz}||_2 \le (1+\varepsilon)||\mathbf{z}||_2\} \ge 1-\delta$ for any $0<\varepsilon$ and $\delta<1/2$ when $d=O(\min\{D,\varepsilon^{-2}\log(1/\delta)\})$. (D)JL lemmas are often formulated in terms of squares of norms and distances.

The proof of DJL lemmas is based on the concentration (for concrete classes of $\mathbf{R}$) of $||\mathbf{Rz}||_2$ around $||\mathbf{z}||_2$ (or $||\mathbf{Rz}||_2^2$ around $||\mathbf{z}||_2^2$). For the DJL lemma, the value of $d=\Omega(\varepsilon^{-2}\log(1/\delta))$ is the dense lower bound [12, 13].

The truth of the JL lemma is obtained with high probability from the DJL lemma with the use of Boole's inequality (union bound) by choosing $\delta < 2 / N / (N - 1)$ and putting $\mathbf{z} = \mathbf{x} - \mathbf{y}$ for all $N(N-1)/2$ pairs of vectors. Note that the value of $1 \pm \varepsilon$ in the DJL lemma is independent of $|| \mathbf{z} ||_2$ and in the JL lemma is independent of $|| \mathbf{x} - \mathbf{y} ||_2$. Therefore, in proofs, the norm of these vectors can be considered to be a unit norm.

Random projection provides the obliviousness of a transformation. The dimension $d = \Omega(\varepsilon^{-2} \log N)$ in the JL lemma is the dense lower bound for the linear transformation $\mathbf{Rz}$ [14] and in the general case [15].

We call JLTs classes of matrices of a linear transformation for which the JL lemma holds. A large (but not alone) JLT class consists of matrices with i.i.d. elements, i.e., independently and identically distributed random quantities (r.q.) from a sub-Gaussian distribution [11, 16–18]. A centered r.q. x is sub-Gaussian if $\exists c > 0 \ \forall \lambda > 0 \ \Pr\{|x| > \lambda\} \le 2\exp(-c\lambda^2)$. For example, sub-Gaussian JLTs are matrices with i.i.d. elements from the Gaussian distribution $\mathrm{Norm}(0, 1)$ with binary elements from $\{-1, +1\}$ with probability 1/2 (the Rademacher distribution), with ternary elements from $\{-1/q^{1/2}, 0, +1/q^{1/2}\}$ with corresponding probabilities $\{q/2, 1-q, q/2\}$, etc. [11, 16–18]. The dimension d in (D)JL lemmas depends on c. Note that, for the truth of the DJL lemma, $\mathbf{Rz}$ with such $\mathbf{R}$ should be multiplied by $1/\sqrt{d}$ or $\mathbf{R}$ should be obtained a result of the same scaling of (sub-Gaussian) r.q.

Let us consider the connection of JLTs with matrices that have the following restricted isometry property (RIP) [18–20]: for any k-sparse vector (i.e., a vector with no more than k nonzero components), the multiplication by a RIP matrix preserves the square of the Euclidean norm with distortion $1 \pm \varepsilon$. Such RIP matrices are used in problems of compressed sensing (see references in [18–20]). For JLT matrices, RIP is satisfied with high probability (with other constants and for vectors with sparseness k up to some optimal). For example, Gaussian and Rademacher random matrices $\mathbf{R}$ satisfy RIP when $d = O(\varepsilon^{-2} k \log(D/k))$ [20] and, vice versa, a matrix $\mathbf{RD}_R$ (where $\mathbf{R}$ is a $\mathrm{RIP}(k, \varepsilon/4)$-matrix and $\mathbf{D}_R$ is a diagonal Rademacher matrix) is a JLT with high probability (with a suboptimal d, $d = O(\varepsilon^{-2} k \log D)$ when $N \le 2^k$) [21, 22]. Recent results on RIP are given in [20–23].

In addition to RIP, analogues of the JL lemma also exist with some constraints for other infinite (continuous) sets such as manifolds, linear subspaces, unions of subspaces, etc. (see [18, 24] and references to them). In particular, let a continuous set S have a Gaussian mean width ω that is defined as $\omega(S) = \mathrm{E}\{\sup_{x \in S} \langle \mathbf{r}, \mathbf{x} \rangle\}$, where $\mathbf{r} \sim \mathrm{Norm}(\mathbf{0}, \mathbf{I}_D)$ and E is mathematical expectation (m.e.). For a Gaussian i.i.d. $\mathbf{R}$, when $d = O(\omega^2(S)/\varepsilon^2)$, an analogue of the JL lemma with additive distortion $\pm\varepsilon$ (3) [24–27] holds for S. For matrices with sub-Gaussian i.i.d. elements, a similar result is obtained in [28] (see also [18]). Note that $\omega(S) \le (2 \log |S|)^{1/2}$ [27].

In multiplying RIP matrices by $\mathbf{D}_R$, for different distortion ε and sparseness k levels, an additive analogue of the JL lemma holds with high probability for continuous bounded sets S when d depends on $\omega^2(S)$ [27]. The existence of RIP matrices with fast multiplication (Sec. 3) allows one to accelerate this random projection.

2.2. Derandomization of random projection. The conditions of the JL lemma presume the need for randomized embeddings for dimensionality reduction since, when $d < D$, for a deterministic matrix, there are infinitely many $\mathbf{x}$: $\mathbf{Rx} = 0$ (vectors from the null-space of $\mathbf{R}$). Therefore, the derandomization of JLT consists in searching for JLT matrix classes that can be generated (or can be chosen from all "ready" matrices of some distribution) with a minimal number of random bits (see [13, 29, 30] and references in them). It is important for applications with limited memory. In particular, matrix generation in [30] requires only $d = O(\log(1/\delta) \log D)$ random bits (Sec. 3.3). However, there are procedures for constructing explicit JLT matrices for a given set of N vectors [31, 32].

2.3. Dispersions of estimates. Analogues of the JL lemma for the dot product, angles, and embeddings into L_1. The JL lemma provides probabilistic guarantees of the worst-case $1 \pm \varepsilon$-distortion of the Euclidean distance. The dispersion V of an estimate for a distance/similarity measure is the measure of inaccuracy of embedding on the average (and, in some cases, makes it possible to obtain estimates for the worst case when error distribution is known). The randomness of estimates in random projection is conditioned by different implementations of $\mathbf{R}$. The dispersion V of the estimate for $|| \mathbf{x} - \mathbf{y} ||_2^2$ on the basis of d-dimensional sketches [33, 34] is calculated as follows:

$$\mathrm{V}\{|| \mathbf{x} - \mathbf{y} ||_2^{2*}\} = 1/d \left((\mathrm{E}\{\rho^4\} / \mathrm{E}^2\{\rho^2\} - 3) \sum_{i=1}^{D} (x_i - y_i)^4 + 2|| \mathbf{x} - \mathbf{y} ||_2^4 \right), \tag{4}$$

where ρ is an r.q. (an element of **R**). For ternary matrices with i.i.d. elements from $\{-1/q^{1/2}, 0, +1/q^{1/2}\}$ with probabilities $\{q/2, 1-q, q/2\}$, $\mathrm{E}\{\rho^4\}/\mathrm{E}^2\{\rho^2\}=1/q$ [33], and, for binary matrices with elements from $\{0,1\}$: $1/(q-q^2)-3$ [34]. For Gaussian i.i.d. matrices, V gives formula (4) with $\mathrm{E}\{\rho^4\}/\mathrm{E}^2\{\rho^2\}=3$ [35, 33].

To estimate the dot product $\langle \mathbf{x}, \mathbf{y} \rangle$ [33, 34], we have

$$\mathrm{V}\{\langle \mathbf{x}, \mathbf{y} \rangle^*\} = 1/d\left((\mathrm{E}\{\rho^4\}/\mathrm{E}^2\{\rho^2\}-3)\sum_{i=1}^{D} x_i^2 y_i^2 + \langle \mathbf{x}, \mathbf{y} \rangle^2 + \|\mathbf{x}\|_2^2 \, \|\mathbf{y}\|_2^2 \right). \qquad (5)$$

The accuracy of estimates [33] can be increased by taking into account the values of norms $\|\mathbf{x}\|_2$ and $\|\mathbf{y}\|_2$.

The equality $\|\mathbf{x}-\mathbf{y}\|_2^2 = \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 - 2\langle \mathbf{x}, \mathbf{y} \rangle$ implies the existence of analogues of (D)JL lemmas for JLT matrices for $\langle \mathbf{x}, \mathbf{y} \rangle$ and the cosine of the angle $\cos(\mathbf{x}, \mathbf{y})$ between the unit $\mathbf{x}$ and $\mathbf{y}$. However, in this case, a distortion $\pm\varepsilon_a$ is additive distortion (3) and, at the same time, $\varepsilon_a = \varepsilon \|\mathbf{x}\|_2 \|\mathbf{y}\|_2$ [36, 29, 37]. A lemma with optimal probabilities δ (as in the JL lemma) is presented in [37], where it is also shown that, for a $1\pm\varepsilon$-distortion, δ depends on $\cos^2(\mathbf{x}, \mathbf{y})$. Fast estimation of $\langle \mathbf{x}, \mathbf{y} \rangle$ can be useful in estimating kernel similarities (Sec. 7).

For transformations preserving Euclidean distances with $1\pm\varepsilon$-distortion, there is an analogue of JL lemmas for estimating the angle between unit vectors with $\pm\varepsilon_a$-distortion [29, 38]. The preservation of an angle with distortion $1\pm\varepsilon$ requires the increase in the number of preserved distances, i.e., in dimension d (and/or ε) [39].

To embed N vectors from L_2 into L_1, the conditions (of analogues) of (D)JL lemmas with distortion $1\pm\varepsilon$ are fulfilled with insignificant changes in constants (for a Gaussian random i.i.d. matrix [26, 40]). The use of sparse matrices (with a small fraction of nonzero elements) requires the "smoothness" of $\mathbf{x}$ [41, 11] (for sparse Gaussian matrices [41, 11], for ternary matrices [11], and for embeddings into L_2, see Sec. 3.1).

The embedding of continuous bounded sets with $\omega(S)$ from L_2 into L_1 with the help of a Gaussian matrix is possible with distortion $\pm\varepsilon_a$ for $d = O(\omega^2(S)/\varepsilon_a^2)$ [26, 42], and the embedding of the entire D-dimensional space L_2^D into L_1^d with distortion $1\pm\varepsilon$ is possible only for $d = O(D)$ [3, 4]. The results testifying to the impossibility of dimensionality reduction for L_s, $s \neq 2$, are given in Sec. 4. Advantages and drawbacks of embedding vectors of Euclidean spaces are considered Sec. 9.1.

3. ACCELERATION OF RANDOM PROJECTION

The drawbacks of JLT random projection include a large computational complexity $O(Dd)$ of multiplication of a vector by a matrix in the case of direct implementation (but see [43]). Time decreases up to $O(\mathrm{nnz}(\mathbf{x})d)$ for a sparse $\mathbf{x}$ with the number of nonzero components $\mathrm{nnz}(\mathbf{x})$. For arbitrary **x**, some acceleration of JLT can be achieved by the use of special sparse or nonsparse matrices making it possible to perform high-speed multiplication.

3.1. Sparse random i.i.d. matrices. The time of multiplying a dense **x** by traversing along nonzero matrix elements (with a fraction or probability q of nonzero elements) amounts to $O(Dqd)$, and that for a k-sparse **x** amounts to $O(kqd)$ (see, for example, [44]). Manipulation with matrices consisting of the elements $\{-1, 0, +1\}$ [33, 45–47] and $\{0, +1\}$ is especially efficient [34, 48]. Dispersions of an estimate for the Euclidean distance and dot product from [34] are given by formulas (4) and (5). In [33], the rate of their convergence to the dispersion of Gaussian projections and, in [33, 48], the rate of convergence of the distribution of elements of the output vector to a Gaussian vector are given.

For sparse matrices and sparse vectors **x**, the number of nonzero products $x_i r_i$ (in computing $\langle \mathbf{x}, \mathbf{r} \rangle$ in **Rx**) can be insufficient for the concentration (necessary in the DJL lemma (see Sec. 2.1)) of $\|\mathbf{Rx}\|_2$ around $\|\mathbf{x}\|_2$. For example, a vector with $\|\mathbf{x}\|_2 = 1$ can contain only one single component. Therefore, the DJL lemma for sparse random i.i.d. matrices [41,11] requires to constrain the sparseness of vectors (which is implicitly specified as $\|\mathbf{x}\|_\infty / \|\mathbf{x}\|_2 \le \alpha$, $\alpha \in [1/D^{1/2}, 1]$, and is close to $1/D^{1/2}$) and matrices ($q = C_0 \alpha^2 \log(D/\varepsilon\delta)$). In this case, $d = C\varepsilon^{-2}\log(4/\delta)$ and the m.e. of the number c of nonzero components in a column $\mathrm{E}\{c\} = \tilde{\Omega}(\alpha^2/\varepsilon^2)$, where $\tilde{\Omega}(f)$ designates a function of the form $f \log^{\Omega(1)}(f)$.

3.2. Matrix pipelines for fast JL transform. To accelerate JLT (with the possibility of applying it to sparse vectors), different matrix pipelines are used; a pipeline consists of a sequence of matrices whose multiplication by a vector is rapidly computable, and, for resultant vectors, the JL lemma holds with parameters close to optimal.

The matrix pipeline from [41] is called the fast Johnson–Lindenstrauss transform (FJLT). To achieve the necessary α of a vector **x**, it is preconditioned. In [41], its preconditioning is provided by a random rotation of **x** by means of $\mathbf{HD}_R\mathbf{x}$, where **H** is an (orthogonal) Hadamard matrix,

$$\mathbf{H}_1 = (1),\ \mathbf{H}_D = \frac{1}{\sqrt{2}}\begin{pmatrix} \mathbf{H}_{D/2} & \mathbf{H}_{D/2} \\ \mathbf{H}_{D/2} & -\mathbf{H}_{D/2} \end{pmatrix}.$$

A vector is multiplied by **H** in time $O(D \log D)$. At the same time, $\alpha = O(\sqrt{d/D}) \sim O(\sqrt{\log N/D})$ is reached with high probability, which allows one to use a sparse i.i.d. matrix **G** (in [41], a Gaussian matrix with an optimal d and with $q \sim d^2/D$ and $\mathrm{nnz}(\mathbf{G}) = O(d^3)$). As a result, the FJLT transformation $\mathbf{GHD}_R\mathbf{x}$ is obtained. The execution time amounts to $O(D \log D + d^3)$. A further development [49–51] and application of RIP matrices (see Sec. 2.1) made it possible to decrease and then to eliminate the dependence of time on d owing to increasing d in comparison with the time optimal in the JL lemma. In particular, in [20], the execution time equal to $O(D \log D)$ is reached for **x** with $\mathrm{nnz}(\mathbf{x}) \le D/\mathrm{poly}\log D$ when $d = O(\varepsilon^{-2} \log N \log^3 D)$.

Very simple matrices providing the multiplication time of order of $O(D \log D)$ are Toeplitz and circulant matrices. Toeplitz matrices have identical elements on their diagonals (specified by $D + d - 1$ numbers). In circulant matrices (we denote them by **C**) rows are obtained by a cyclic shift of the first row (i.e., D numbers are required). For random projection, elements of a row are usually Gaussian or Rademacher i.i.d. r.q. ($\mathrm{vec}(\mathbf{D}_G)$ or $\mathrm{vec}(\mathbf{D}_R)$). With high probability, such matrices are RIP matrices [52] if parameters are appropriately chosen, i.e., $\mathbf{CD}_R$ is a JLT (see Sec. 2.1 and [21]). An improvement in the analysis of the JLT pipeline $\mathbf{CD}_R$ made it possible to improve the required d from $O(\varepsilon^{-2} \log^3 N)$ [53] to $O(\varepsilon^{-2} \log^2 N)$ [54] and even to $O(\varepsilon^{-2} \log^{(1+\eta)} N)$ (see [55, 56] and references from them).

Similar fast pipelines are also used for fast implementation of RIP transformations (see Sec. 2.1) in obtaining binary sketches [1] and approximation of kernels (Sec. 7) and linear parts of layers of neural networks [57–60]. Note that elements of the matrix of the product of matrices of a pipeline are not (Gaussian) i.i.d., which complicates the analysis of such "structured" random matrices.

3.3. Sparse JL transformation. In all FJLT versions, a possible sparseness of **x** is not used (and vice versa, compacting is often performed). As a result, for a vector with one nonzero component (for example, in the mode of streaming processing (see Sec. 1.3)), the time $O(D \log D)$ of sketch modification is much larger than the "naive" $O(d)$. Moreover, in many applications, sparse vectors are used (representations of texts by words, recommendations or purchases of users, etc.).

To accelerate the multiplication of sparse vectors by sparse matrices and to overcome the constraint on c (see Sec. 3.1), i.i.d. r.q. are not used as matrix elements. In particular, in [61], the following so-called the hashing trick is proposed: an unbiased estimate for dot product is found from sketches obtained using hash functions $h:[D] \to [d]$ and $g:[D] \to \{-1, +1\}$, where $[n]$ denotes $\{1, 2, \dots, n\}$. A sketch component is formed by adding the components of the initial vector that are mapped into the component and are multiplied by values from $\{-1, +1\}$ that correspond to them. This is identical to the multiplication by a matrix with exactly one (randomly located) +1 or −1 in a column. In this case, the dispersion of an estimate is the same as for matrices with i.i.d. r.q. taken from $\{-1, +1\}$ [61, 62]. To achieve the necessary α, a simple deterministic compacting of **x** is used with the help of c-fold "reproduction" of its components and division of them by $c^{1/2}$, which provides the preservation of $\|x\|_2$ and reduction in $\|x\|_\infty$ by a factor of $c^{1/2}$. In this case, hashing is modified as $h:[cD] \to [d]$ and $g:[cD] \to \{-1, +1\}$. The resultant transformation of **x** can be implemented by multiplying by a pseudo-random matrix of size $d \times D$; the number of nonzero elements in columns of this matrix varies from one to c (owing to possible collisions between c hashes of the same component **x**).

As a result of analysis of the considered scheme as a JLT [63], $c = O(\varepsilon^{-1} \log(1/\delta) \log^2(d/\delta))$ is obtained for $d = O(\varepsilon^{-2} \log(1/\delta))$. As is shown in [64], it suffices that $c = O(\varepsilon^{-1} \log(1/\delta) \log(d/\delta))$; some improvement in this parameter is attained in [65]. To further reduce c, it is proposed [30] to use exactly c nonzero elements from $\{-1, +1\}$ in a column of the matrix $\mathbf{R}$. In one of versions, a column of the matrix $\mathbf{R}$ is divided into c continuous blocks of dimension d / c and one element -1 or $+1$ is randomly placed in each of them. This makes it possible to improve sparseness up to $c = \Theta(\varepsilon^{-1} \log (1/\delta)) \sim \varepsilon^{-1} \log|S|$ with optimal $d = O(\varepsilon^{-2} \log (1/\delta))$. Thus, the fraction of nonzero matrix elements and computational speedup of this sparse JL transformation (SJLT) are equal to ε. This c is close to the optimal $c = \Omega(\varepsilon^{-1} \log (1/\delta) / \log (1/\varepsilon))$ [66].

In [24], the truth of analogues of the JL lemma is investigated with distortion $1 \pm \varepsilon$ of the Euclidean distance for SJLT and different sets of unit vectors under some constraints on geometry by means of a "complexity parameter."

Thus, structured (i.e., not Gaussian i.i.d.) matrices make it possible to accelerate multiplication, to reduce memory expenditures for storing matrices and the number of required random numbers, and to simplify algorithmic implementations. A comparative experimental investigation of dimensionality reduction algorithms on the basis of the JL lemma for different matrix pipelines is given in [67].

4. EMBEDDINGS AND SKETCHES FOR ESTIMATING NON-EUCLIDEAN MINKOWSKI DISTANCES

For other distances L_s, $s \neq 2$, a dimensionality reduction $L_s^D \to L_s^d$ by a linear transformation with a distortion (constant for vectors of length d and independent of D) is impossible in the general (worst) case (for L_1, see [68, 69]). Proofs are based on the demonstration of collections of N vectors in a space whose dimension is $D = N$ and whose embedding with a given distortion requires a high dimension.

In particular, for a linear embedding of N vectors from L_s into L_s^d when $1 \leq s \leq \infty$, the multiplicative distortion A (2) is no less than $A = \Omega((N/d)^{|1/s-1/2|})$ [70]. For L_1, this means that $d \geq CN / A^2$. This is also true for the embedding of L_1 into any L_s [70]. A stronger result for L_1 asserts that $d \geq N^{\Omega(1/A^2)}$ [69, 68]. For small distortions $1+\varepsilon$, the lower bound $d \geq N^{1-O(1/\log(1/\varepsilon))}$ [71] and the upper bound $d \leq O(N/\varepsilon^2)$ [72].

However, a transformation of a metric without dimensionality reduction can also be useful (see Sec. 1.3). Note that, for $1 \leq t \leq s \leq 2$, the entire space L_s^D can be embedded into L_t^{CD} with distortion $1+\varepsilon$ and, in this case, $C = C(s, t, \varepsilon) = O(\varepsilon^{-2} \log(1/\varepsilon))$ is not too large [73, 74, 3, 4] (and see also Sec. 3.1 and [4] for $L_2 \to L_1$). When $C > 1$, an explicit embedding can be constructed. However, characteristics of such embeddings are much worse than those of randomized ones [4].

Thus, for L_s, $s \neq 2$, it is impossible to linearly reduce the dimensionality of an arbitrary set of N vectors with constant distortion in the worst case. To overcome this restriction, constraints on sets of vectors (Sec. 4.1) and sketches with estimation of L_s ($0 < s \leq 2$) not in terms of distance in the target space (Sec. 4.2) are used. Note that, for operating with distances L_s when $s > 2$, efficient sketches [75] (of constant dimension and with constant distortion independent of D) are impossible in principle, and the required dimension $d = \Omega(D^{1-2/s})$ [76].

4.1. Embeddings of subsets of L_1. For k-sparse vectors from L_1, $1 \pm \varepsilon$-embedding with $d \geq Ck \log(D/k)/\varepsilon^2$ is possible (with high probability) with the help of L_1-RIP matrices (in particular, scaled binary matrices containing $C\varepsilon^{-1} \log(D/k)$ 1s in each column) (see [77] and Prop. 1 in [78]). Similar results for $1 \leq s \leq \infty$ are obtained in [79].

The approach to the embedding of s-block norms [78] into L_1 uses the element-wise multiplication of matrices one of which is binary with a fixed number of 1s randomly located in each column and the other is Gaussian. By varying s, it is possible to reproduce well-known results for embeddings $L_2 \to L_1$ and $L_1 \to L_1$ for subsets of vectors with certain properties.

Nonlinear embeddings from L_s into L_t, $1 \leq t \leq s \leq 2$, with distortion $1+\varepsilon$ in a bounded range of distances with target dimension $d = O(\log N)$ that also depends on values of the range and on ε, t, and s, are proposed in [80]. A one-dimensional embedding of $\mathbf{x}$ it is performed as $\sin(2\langle \mathbf{x}, \mathbf{r} \rangle / a + \xi)$ with scaling, where $\mathbf{r}$ is a random vector from an s-stable distribution (Sec. 4.2), a is a value connected with the range size, and $\xi \sim \mathrm{Unif}[0, 2\pi]$. Such embeddings (with a bounded range) are useful in similarity search, clusterization, etc.

4.2. Sketches for distances L_s ($0< s \leq 2$) on the basis of stable random projections. One more approach to the fast estimation of Minkowski distances L_s ($0< s \leq 2$) with small distortion consists of creating sketches of small dimension $d = O(\log N)$ with which, as opposed to embeddings, other estimates of initial distances [81] are used instead of L_s. For them, there are analogues of the JL lemma with distortion $1 \pm \varepsilon$ for estimates for distances computed from sketches.

Sketches for L_s ($0< s \leq 2$) are formed on the basis of stable random projections as **Rx** (**R** is a random matrix with i.i.d. elements from an *s*-stable distribution [81]). For example, the 1-stable Cauchy distribution is used for L_1. Note that the Gaussian distribution is 2-stable.

To estimate L_s from sketches, versions of median estimators for absolute values of the difference between components of sketches are used [81]. To increase accuracy, median estimators, geometric mean estimators, harmonic mean estimators, fractional power estimators, optimal quantile estimators, and maximum likelihood estimators (with bias correction) are used (see [82, 83] and references in them). Advantages and drawbacks of sketches obtained by a stable random projection are described in Sec. 9.2.

5. ESTIMATION OF LINEAR SUMMARY STATISTICS FROM SKETCHES OBTAINED BY SAMPLING

The selection of a subset of elements from the initial representation of an object is called sampling. The objective of sampling usually is the obtaining of a representation of the object that makes is possible to estimate some of its characteristics. To rapidly estimate distance/similarity measures with the use of sampling, representations of initial objects that can be considered as vectors are mainly used. Sketches obtained by sampling can also be represented by vectors. (The representation in the form of pairs (ID, value) with value $\neq 0$ is often used in which components with an identical ID correspond to one another. Such representations can be easily converted into usual vectors.)

We now consider methods of random sampling (see [84, 85] and references in them). In simple random sampling with replacement, each component is selected for a sketch with equal probability (and it can be selected several times). The number of sample components selected for a sketch is fixed. In PPS (probability proportional to size) sampling with replacement, the probability of selection is proportional to the weight of a component (for example, its size). For data with heavy tails (where the main part of weight is concentrated in a small number of components with large values), as a result of such sampling, a sketch can consist of the same heavy components, and the others will be poorly presented.

In sampling without replacement, a component is selected no more than once, which allows one to obtain more exact estimates. It is possible to distinguish between (Bernoullian) sampling with equal probability and with different probability that is, for example, proportional to component sizes (Poisson sampling). Both types of sampling select an unfixed number of components. If this number is fixed, then Bernoullian sampling becomes simple random sampling without replacement and Poisson sampling becomes conditional sampling.

Bernoullian sampling is analyzed more easily, but estimates have a lower accuracy for data with heavy tails. Poisson sampling allows one to find more exact estimates, but their obtainment from sketches is a nontrivial task; constraints on data are also used, for example, the nonnegativity of vector components, which corresponds to weighted sets.

A random sampling of the representation of an object does not take into account information on other objects (i.e., it is oblivious). However, in sampling different objects, both different collections of random numbers (independent sampling) and identical ones (coordinated sampling) can be used. To estimate similarity, coordinated sampling is mainly used. Dispersions of estimates of some quantity or other that are obtained from sketches by sampling decrease with increasing the sketch dimension d.

5.1. Sketches obtained by simple random sampling without replacement. In simple random sampling without replacement, sketches of dimension d are obtained by a random permutation of initial vectors (to eliminate a structure potentially existing in them) and selection of their first d components. Note that the same sketch is obtained as a result of multiplying an input vector by the corresponding binary matrix with one 1 in each row.

Let the distance/similarity measure sim of initial vectors be defined as $\mathrm{sim}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{D} \mathrm{sim}_i(x_i, y_i)$, i.e., is linear summary statistics, for example, the dot product, squared Euclidean distance, χ^2 distance (Sec. 8.1), etc. Then an unbiased estimate $\mathrm{sim}^*(\mathbf{x}, \mathbf{y})$ is obtained from the value of $\mathrm{sim}(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ computed from sketches $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ of dimension d and its dispersion V is obtained [86] as follows:

$$\mathrm{sim}^*(\mathbf{x}, \mathbf{y}) = \mathrm{sim}\,(\hat{\mathbf{x}}, \hat{\mathbf{y}}) D / d, \tag{6}$$

$$\mathrm{V}\{\mathrm{sim}^*(\mathbf{x}, \mathbf{y})\} = (D/d)(D-d)/(D-1)\left[\sum_{i=1}^{D} \mathrm{sim}_i^2(x_i, y_i) - \mathrm{sim}^2(\mathbf{x}, \mathbf{y})/D\right]. \tag{7}$$

For $D >> d$ and also for $\sum_{i=1}^{D} \mathrm{sim}_i^2(x_i, y_i) >> \mathrm{sim}^2(\mathbf{x}, \mathbf{y})/D$ (i.e., for vectors with heavy tails), the value of V is large.

5.2. Sketches obtained by conditional random sampling for sparse vectors. For strongly sparse vectors, specialized sampling methods yield more exact estimates of initial distance/similarity measures with the same memory capacity per sketch. Conditional random sampling (CRS) [86] forms a sketch by selecting a given number of first nonzero components of a vector after a random permutation (zero components are not used in the sketch). In computing estimate (6), $d = \min\{\max \mathrm{ID}(\hat{\mathbf{x}}) - 1, \max \mathrm{ID}(\hat{\mathbf{y}}) - 1\}$ is used [86], where $\max \mathrm{ID}(\hat{\mathbf{x}})$ is the maximal ID component in a sketch $\hat{\mathbf{x}}$. If $d_{\hat{\mathbf{x}}}$ is the number of components in the sketch $\hat{\mathbf{x}}$, then the dispersion

$$\mathrm{V}\{\mathrm{sim}^*(\mathbf{x}, \mathbf{y})\} \approx D/(D-1)(\max\{\mathrm{nnz}(\mathbf{x})/(d_{\hat{\mathbf{x}}} - 1), \mathrm{nnz}(\mathbf{y})/(d_{\hat{\mathbf{y}}} - 1)\} - 1)$$
$$\times \left[\sum_{i=1}^{D} \mathrm{sim}_i^2(x_i, y_i) - \mathrm{sim}^2(\mathbf{x}, \mathbf{y})/D\right].$$

Comparing this dispersion with dispersion (7), we see that, as opposed to the use of conventional sampling, this dispersion is approximately $D/\mathrm{nnz}(\mathbf{x})$ times less than the latter. For data with heavy tails, dispersion remains large due to the possibility of skipping "heavy" components.

5.3. Sketches obtained by weighted sampling. For data with heavy tails, sampling (without replacement) must give priority vector components of with large values. In (sequential Poisson) priority sampling [84], a sketch of fixed dimension d for vectors with positive components (denoted below by $\mathbf{x} > 0$) is formed as follows. A priority $\beta_i = x_i / r_i$, $i \in [D]$, is assigned to each of its component, where $r_i \sim \mathrm{Unif}(0,1]$ (in the algorithm, they are obtained by hashing the number of the component [87]), and β_i are ordered according to their magnitudes. Then a threshold $\beta = \beta_{d+1}$ is determined. The sketch is formed as $\hat{x}_i = \max\{x_i, \beta\}$ if $\beta_i > \beta$ and 0 otherwise. Note that the priority sampling is suitable for streaming processing and was initially used to estimate sums of components of a vector. In [87], with the help of a special version of such coordinated sketches, the estimate $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y})$ (the generalized Jaccard coefficient for $\mathbf{x}, \mathbf{y} > 0$) [88, 89] is considered,

$$\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{D} \min(x_i, y_i) / \sum_{i=1}^{D} \max(x_i, y_i).$$

It is shown that, for large d, the use of only 2-independent hash functions allows one to reach a small bias (and dispersion) of the estimate $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y})$. Note that $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y})$ is connected with $\|\mathbf{x} - \mathbf{y}\|_1$ [89] through

$$\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y}) = (\|\mathbf{x}\|_1 + \|\mathbf{y}\|_1 - \|\mathbf{x} - \mathbf{y}\|_1) / (\|\mathbf{x}\|_1 + \|\mathbf{y}\|_1 + \|\mathbf{x} - \mathbf{y}\|_1)$$

and the generalization of $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y})$ to real vectors [90] by replacing $x < 0$ by the pair of components $[0\ {-x}]$ and $x > 0$ by the pair $[x\ 0]$.

Sketches obtained by sampling with a view to estimating distances L_s (in particular, L_1 and L_2) between nonnegative vectors are considered in [85]. In addition to priority sampling, Poisson PPS sampling is used for the formation of sketches in which a component is included into a sketch if $x_i > r_i \beta$. The value of β is specified or selected using $\mathrm{E}\{d\} = \sum_{i=1}^{D} \min(1, r_i / \beta)$. Two types of sampling are considered, namely, independent (r_i are different for different sketches) and coordinated (r_i are the same for different sketches). To estimate $(L_s)^s$, in addition to the selected components $\{i, x_i\}$ with $\beta_i > \beta$, $i \in [D]$, their r_i and β_{d+1}, β_d are used. An estimate is obtained from the corresponding components of sketches, depends on s in $(L_s)^s$ and on the type of sampling, and is nontrivial [85]. A general approach to the estimation of other distance/similarity measures from sketches obtained from the results of weighted sampling is presented in [91].

Advantages and drawbacks of sketches obtained by sampling are described in Sec. 9.3.

6. ESTIMATION OF DISTANCES BETWEEN NON-VECTOR DATA

In Sec. 6.1, universal embedding methods for estimating any initial distances are considered, and, in Sec. 6.2, specialized embedding methods for some non-vector distances are described.

6.1. Formation of vector representations on the basis of distances. Methods for formation of vector representations of objects on the basis of their distances to some singled out ("reference") objects (ROs) are universal since they do not require access to initial representations of objects and are applicable to different initial distances. Therefore, spaces of initial representations of objects can be vector, metric, and nonmetric.

In the classical multidimensional scaling (MDS) method [3], an (initial) $N \times N$ matrix dist of distances between a (sub)set of ROs of the base are subjected to "double centering" and are transformed into a similarity matrix $\mathbf{K}$, $K_{ij} = \mathrm{dist}^2(x_0, x_i) + \mathrm{dist}^2(x_0, x_j) - \mathrm{dist}^2(x_i, x_j)$, $i, j \in [N]$. If distances in the initial matrix are Euclidean, then $\mathbf{K}$ is PSD and can be considered as a kernel matrix (Sec. 7). Then vector representations of objects are formed from $\mathbf{K}$ with the help of PCA. This embedding is isometry for N initial objects. This MDS is (weakly) oblivious since approximate embedding of a new object x is also possible by the Nystrom method [92] as $\boldsymbol{\psi}^*(x) = \boldsymbol{\Lambda}^{-1/2} \mathbf{U}^{\mathrm{T}} \boldsymbol{\kappa}_x$, where $\mathbf{U}$ is the matrix of eigenvectors (in columns) obtained as a result of PCA (eigenvalue decomposition of the kernel matrix $\mathbf{K}$ for N objects), $\boldsymbol{\Lambda}^{-1/2} = \mathrm{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots)$, and $\boldsymbol{\kappa}_x = (\kappa(x, x_1), \ldots, \kappa(x, x_N))^{\mathrm{T}}$ are values of kernel similarities of the new object with ROs that can be obtained from the vector of corresponding distances.

Examples of techniques for obtaining approximate oblivious embeddings based on distances are FastMap, MetricMap, and SparseMap (see [93] and references in it). In [94], the FastMap and MetricMap embedding techniques are considered as belonging to the class MDS and using versions or generalizations of the Nystrom method with a loss of accuracy.

In the Frechet isometric embedding [3] from a finite metric space with N objects and dist into L_∞, the ith coordinate of the target vector N-dimensional space is defined as the distance of an object y (one of ROs) to the ith RO: $\mathrm{f}_i(y) = \mathrm{dist}(y, x_i)$, $i \in [N]$. A Frechet isometric embedding is not oblivious.

Contractive embeddings that do not increase initial distances are important for similarity search since they make it possible to obtain exact search results [93]. For example, a distance L_s is contractive for L_t when $s > t > 0$ (without changing vector representations), and also the Frechet embedding is contractive for a new object and L_∞ (or with another distance in L_s in the case of appropriate normalization [93]).

The use of distance-based vector representations for classification and other pattern recognition problems is described in [95]. In [96], for similarity search, vector representations of an object are used whose components are numbers of ROs ordered according to the values of similarities/distances to the object.

Drawbacks of these methods frequently are the heuristic character of selecting ROs, complexity of computation of complicated initial distances (for example, edit distances for graphs [97]), and lack of analytical estimates for distortions.

6.2. Embeddings of objects with special metrics. The design of fast and oblivious algorithms for forming vectors to estimate distances between non-vector initial objects with a specified and minimal distortion is a complicated problem. Therefore, such algorithms are usually specialized for concrete initial representations and types of distances.

Despite the distortion, dimension, and time of obtaining that increase with increasing dimensions of initial representations, embeddings of specialized metrics (distances between non-vector initial objects) into L_1 and also in $(L_2)^2$, L_∞, etc. are needed [75, 98]. This is determined by the existence of efficient sketches for L_s (see Sec. 4 and, for L_1, see Sec. 4.2). Thus, specialized distances between initial non-vector objects are embedded, for example, into L_1 that can be estimated from a sketch of small dimension with a small additional distortion (see Sec. 4.2). Moreover, algorithms for fast similarity search are developed for vectors (Sec. 8.3).

For character sequences (strings), the Levenstein edit distance ($\text{dist}_{\text{edit}}$) [99, 100] is often used; this distance is equal to the minimal number of elementary operations of editing symbols of a string that are necessary for transforming one string into another. Elementary operations are the insertion, elimination, and replacement of a symbol at a certain position. The complexity of computation with the use of dynamic programming is quadratically dependent on the length n of a string.

Embeddings $\text{dist}_{\text{edit}}$ into L_1 mainly operate with strings $\{0,1\}^n$ and use some (continuous) substrings of initial strings as components of sketches , i.e., are nonlinear. Note that, for an alphabet of size 4 [101] and even 2 [102], it is impossible to exactly compute $\text{dist}_{\text{edit}}$ in time $O(n^{2-\varepsilon})$ if the strong exponential time hypothesis [101] holds.

The multiplicative distortion A (2) was analyzed. For the version of $\text{dist}_{\text{edit}}$ with an additional possibility of moving blocks, embeddings are obtained [103] in almost linear time with distortion $\widetilde{O}(\log n)$. However, for the classical $\text{dist}_{\text{edit}}$, similar results have not been obtained for a long time.

In [104], the embedding of the classical $\text{dist}_{\text{edit}}$ is obtained with distortion $\Omega(n^{1/2})$ and computation time $O(n^{3/2})$, and, in [6], it is obtained with distortion $n^{1/3+o(1)}$ in linear time or with distortion $n^{\varepsilon/3+o(1)}$ in time $O(n^{2-\varepsilon})$ (however, not into L_1 but into a space of strings of smaller length).

The distortion equal to $2^{\widetilde{O}(\sqrt{\log n})}$ for an embedding into L_1 (which is less than n^ε for any $\varepsilon > 0$) is given in [105], but it is not known whether it is possible to compute this embedding into subquadratic time. In [106], the same approximation but in time $n^{1+o(1)}$ is obtained, and, in this case, not only embeddings into L_1 but also other non-oblivious embeddings are used. In [107], the distortion equal to $(\log n)^{O(1/\varepsilon)}$ is achieved with the use of sampling of one of strings but in time $n^{1+\varepsilon}$, which is worse than in [106].

The lower bound $\Omega(\log n)$ on the multiplicative distortion of the edit distance between strings $\{0,1\}^n$ in embedding into L_1 is given in [108], and the lower bound $\Omega(n)$ on sketches obtained by a random linear projection is given in [109].

Embeddings of other distances are described in [3, 98] (see also references in them and to them), but not all of them are oblivious even according to a weakened definition (see Sec. 1.2).

7. KERNEL SIMILARITIES AND THEIR APPROXIMATION

A special form of a similarity function is a kernel function (kernel) $\kappa(x, y)$ [110]. It is a continuous, real-valued, symmetric, and positive semidefinite (PSD) function. One of definitions of the $\kappa(x, y)$ is connected with the existence of a (possibly, implicit) transformation $\boldsymbol{\varphi} : X \to H$ of initial objects x and y into vectors $\boldsymbol{\varphi}(x)$ and $\boldsymbol{\varphi}(y)$ in a (possibly infinite dimensional) "secondary" or "feature" Hilbert space H and this transformation is such that $\kappa(x, y) = \langle \boldsymbol{\varphi}(x), \boldsymbol{\varphi}(y) \rangle$.

Kernel similarity is computed from initial representations of objects of some type (vectors, sequences, graphs, etc.) with the help of a kernel function. The complexity of computation of kernels depends on the concrete type of a kernel and usually is polynomial. Examples of kernels for vectors $\mathbf{x}$, $\mathbf{y}$ are $\kappa(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$, i.e., the linear kernel, polynomial kernel

$$\kappa(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^s, \quad c \geq 0, \tag{8}$$

and Gaussian RBF

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp(-1/2 \, || \, \mathbf{x} - \mathbf{y} \, ||_2^2 / \sigma^2). \tag{9}$$

Other examples are kernels for objects (graphs, etc.) structured on the basis of their partition into substructures with their local kernel similarities [111–114].

Kernel-based algorithms depend only on $\kappa(x, y)$, but usually require the computation (and use of) N^2 kernel similarities between N objects (i.e., the kernel matrix $\mathbf{K}$). For large N, this is often impossible.

For fast estimation of elements of $\mathbf{K}$, a low-rank approximation of $\mathbf{K}$ by the product of matrices of small rank is used; these matrices are obtained using a random projection or a sampling of $\mathbf{K}$ and also pseudo-inverse (versions of the Nystrom method, see Sec. 6.1 and [115–118]). Moreover, for kernels that are functions of distance/similarity values of high-dimensional vectors, the fast estimation of these distances/similarities from sketches/embeddings (see Secs. 2–5) destined for this purpose accelerates kernel estimation. One more approach is the obtaining of vector representations of initial (possibly, non-vector) objects x and y whose dot product makes it possible to exactly or approximately accelerate the computation of $\kappa(x, y)$. In this approach, algorithms can be used that directly operate with vectors, which often turns out to be more efficient than the application of kernel-based algorithms.

The explicit formation of vectors $\boldsymbol{\varphi}(x)$ allows one to directly use them. Examples are polynomial kernel (8) and explicit vector representations for graph kernels [97, 112, 114, 119–121]. The dimensionality reduction of $\boldsymbol{\varphi}(x)$ can be performed by the methods described in Secs. 2–5. Drawbacks include a very high dimension of H in many cases (for example, D^s for polynomial vector kernels (8)) or infinite (for example, for RBF kernels (9)), the complexity (impossibility) of transformation of $\boldsymbol{\varphi}(x)$, and also expenditures for dimensionality reduction.

Let us consider methods of direct formation of vector representations for fast estimation of kernel similarities from representations or similarities of initial objects.

The Nystrom method (see Sec. 6.1) requires ROs (adaptation to data) and forms vector representations preserving similarity but uses a computationally intensive eigenvalue or singular value decomposition.

After the publication of [122], the approach was gaining ground that consisted of the oblivious formation of vector representations for approximating kernels whose representation is known in the form [123]

$$\kappa(x, y) = \mathrm{E}_{\mathbf{w}}\{\psi(x, \mathbf{w})\psi(y, \mathbf{w})\}, \tag{10}$$

where $\mathbf{w}$ is a random vector of parameters from some distribution dependent on κ but not on x and y; $\psi(x, \mathbf{w})$ is a random feature map (RFM) for the kernel κ.

To approximate $\kappa(x, y)$, $\mathbf{w}_i$, $i \in [d]$, are selected from the distribution and $\psi_i = \psi(x, \mathbf{w}_i)$, $i \in [d]$, are computed that are assigned to components of the vector $\boldsymbol{\psi}$. The estimate for the kernel is obtained by the formula $\kappa^*(x, y) = \langle \boldsymbol{\psi}(x), \boldsymbol{\psi}(y)\rangle / d$. An increase in d decreases the dispersion of the estimate.

For shift-invariant kernels $\kappa(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{x} - \mathbf{y})$ (Gaussian, Laplace, Cauchy, etc. RBFs), according to the Bochner theorem [122], there is a representation of the form (10). A component ψ_i is formed by a nonlinear transformation of the value of $\langle \mathbf{x}, \mathbf{w}\rangle$, where $\mathbf{w} \sim p(\mathbf{w})$ and $p(\mathbf{w})$ is the inverse Fourier transform of the kernel κ. For example, for RBF (9), $\psi(x, \mathbf{w}, U) = \sqrt{2}\cos(\langle \mathbf{x}, \mathbf{w}\rangle + U)$, where $\mathbf{w}$ is taken from the Gaussian distribution $\mathrm{Norm}(\mathbf{0}, \mathbf{I}/\sigma^2), U \sim \mathrm{Unif}[0, 2\pi]$.

Note that though the use of vector representations of RFMs shows good results for learning problems concerning linear models, but if there is a difference in the spectrum of kernel eigenvalues, then the vector representations obtained by the Nystrom method yield better results than RFMs [124].

Versions of the mentioned transformation are given in [125]. Additive distortions of approximations of shift-invariant kernels are investigated in [122, 125, 126] (but see [127]). To decrease d with the same distortion, $\mathbf{w}$ is generated using a quasi-Monte Carlo method and also learning [128].

To accelerate random projections in forming RFMs, collections of d / D matrix pipelines $\mathbf{D}_S\mathbf{HD}_G\mathbf{PHD}_R$ with the following matrices are used and analyzed [129]: $\mathbf{P}$ (a random permutation), $\mathbf{D}_G$ (a diagonal Gaussian i.i.d. matrix), and $\mathbf{D}_S$ (a diagonal scaling matrix). The matrix $\mathbf{CD}_R$ is used in [130]. The formation of vectors $\mathbf{w}$ with the use of learning for a more exact approximation of a given kernel and improvement in the quality of classification with a minimal d is investigated in [131, 132].

The Bochner theorem is also applicable to a family of additive homogeneous kernels [133] that are functions of the scalar signature of a kernel and include kernels of intersection $\min\{x, y\}$, Hellinger, χ^2, and Jensen–Shannon (JS) (Sec. 8.1). As distinguished from [122], it is proposed to compute components of a feature vector without random sampling and with the use of explicit analytical expressions.

For kernels $\kappa(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{x}+\mathbf{y})$ with $\mathbf{x}, \mathbf{y} \geq 0$, an extension of the Bochner theorem is used and $p(\mathbf{w})$ is obtained by the inverse Laplace transform, $\psi_i(\mathbf{x}) = \exp(-\langle \mathbf{x}, \mathbf{w}_i \rangle)$ [134].

A function $\kappa(\mathbf{x}, \mathbf{y}) = \mathrm{f}(\langle \mathbf{x}, \mathbf{y} \rangle)$ (for example, kernel (8)) is a PSD kernel if $\kappa(\mathbf{x}, \mathbf{y}) = \mathrm{f}(z)$ is decomposable into a Maclaurin series $\mathrm{f}(z) = \sum_{i=1}^{\infty} a_i z^i$ with $a_i \geq 0$. In [135], a sketch $\psi_i(\mathbf{x}) = (a_N 2^{N+1})^{1/2} \prod_{j=1}^{N} \langle \mathbf{x}, \mathbf{w}_j \rangle$ is proposed, where $\mathbf{w}_j = \mathrm{diag}(\mathbf{D}_R)$, N is a random number, and $\Pr[N = n] = 1/2^{n+1}$. In [136], the fact is used that the tensor product of a vector with itself that is repeated s times yields an embedding into H that corresponds to the kernel $\langle \mathbf{x}, \mathbf{y} \rangle^s$. For each vector, s different sketches of dimension d are created using the hashing trick [61]. The final sketch of dimension d is obtained by the computation (of a d-dimensional vector) of the FFT of each sketch and by the component-wise multiplication of them and execution of FFT^{-1}. The time of obtaining of a sketch amounts to $O(sD + sd \log d)$ and usually equals $d = O(D)$.

In [137], for more compact vectors with a better kernel approximation, high-dimensional vectors are first obtained with the help of transformations from [135] or [136], and then the (F)JLT is applied (see Sec. 3.2). The Bochner theorem is not directly applicable to a polynomial kernel [138], however, for unit vectors [138], it was possible to approximate $p(\mathbf{w})$. This gives a more exact approximation of kernels for large s.

Methods for formation of binary vectors for approximation of kernels are given in [1].

8. OTHER LINES OF INVESTIGATION

8.1. Embeddings of distances between distributions. Statistical (and also probabilistic and information) distances are introduced for vectors with $x_i \geq 0$ and $\sum_{i=1}^{D} x_i = 1$. Such vectors can be considered as distributions or points on a D-dimensional simplex, i.e., a multidimensional generalization of a triangle. Many statistical distances are not metrics and even are asymmetrical. Some of them are called (statistical) divergences.

It is noted in [139] that, for metric distances such as the statistical Hellinger distance $\mathrm{dist}_{\mathrm{Hell}}^2 = 1/2 \,||\, \mathbf{x}^{1/2} - \mathbf{y}^{1/2} ||_2^2$ and Mahalanobis distance $\mathrm{dist}_{\mathrm{Maha}}^2 = (\mathbf{x}-\mathbf{y})^{\mathrm{T}} \mathbf{A} (\mathbf{x}-\mathbf{y}) = ||\mathbf{L}(\mathbf{x}-\mathbf{y})||_2^2$ dimensionality reduction with distortion $1 \pm \varepsilon$ is possible according to the JL lemma since they use the squared Euclidean distance between transformed vectors $\mathbf{x}$ and $\mathbf{y}$. It is shown that, for embeddings of nonmetric distances (divergences) of Bhattacharya $\mathrm{dist}_{\mathrm{Bhat}} = -\ln(\langle \mathbf{x}, \mathbf{y} \rangle)^{1/2}$ and Kullback–Leybler $\mathrm{dist}_{\mathrm{KL}} = \sum_{i=1}^{D} x_i \ln x_i / y_i$ into metric spaces, there are configurations of vectors with an arbitrarily large multiplicative distortion A. For $\mathrm{dist}_{\mathrm{Bhat}}$, the additive analogue of the JL lemma with distortion $\pm \varepsilon_a(\gamma)$ holds when x_i, $y_i \geq \gamma / D$, $i \in [D]$. The analysis is based on the investigation of the distortion in estimating some distance from another for which dimensionality reduction is known according the JL lemma ($\mathrm{dist}_{\mathrm{Bhat}} \to \mathrm{dist}_{\mathrm{Hell}}$ and $\mathrm{dist}_{\mathrm{KL}} \to (L_2)^2$) without a change in vector representations. Note that, in the considered embeddings, vectors in the target space are not located on a simplex.

In [133], explicit representations of $\boldsymbol{\varphi}(\mathbf{x})$ in H are given whose dot product yields values of JS, Hellinger, and χ^2 kernels and also finite-dimensional vector representations for their approximation. This allows one to compute the corresponding divergences of $\mathrm{dist}_{\mathrm{f}}^2 = ||\boldsymbol{\varphi}(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{y})||^2$ and their approximations, but this question is not investigated in [133].

In [140], for $\mathrm{dist}_{\mathrm{Hell}}$, the embedding from a D-dimensional simplex into a d-dimensional one with distortion $1 \pm \varepsilon$ is shown with the help of conventional random projection, but vectors must be located in certain region of the D-simplex and this region decreases with increasing D. In [141], for the JS, Hellinger, and χ^2($\mathrm{dist}_{\chi^2} = \sum_{i=1}^{D} (x_i - y_i)^2 / (x_i + y_i)$) distances and other f-divergences of the class defined in [141], the existence of an analogue of the JL lemma with distortion $1 \pm \varepsilon$ is shown for any N points of a simplex. A nonlinear randomized embedding into $(L_2)^2$ with distortion $1+\varepsilon$ is first performed by analogy with the RFM technique (see Sec. 7) [122]. Then the JL lemma is used for dimensionality reduction to $d = O(\varepsilon^{-2} \log N)$. The obtained vectors are isometrically mapped into an internal region of the simplex and then final points are obtained on the simplex by scaling and centering relative to the simplex centroid. The latter transformation is possible for f-divergences of a certain class [141]. Other results on embeddings of information distances are also given.

Results on the estimation of information distances in streaming models are given in [141, 142].

8.2. On the equivalence of sketches and embeddings. As has been noted in Sec. 4, for L_1 (and for L_s, $0 < s \leq 2$), there are efficient sketches (but not embeddings). Such sketches can be used for initial objects with different representations and distance/similarity measures embedded into L_s (with an insignificant increase in distortion).

However, for embeddings of many specialized metrics into L_s (see Sec. 6.2) with a fixed dimension d of output vectors, distortion increases with increasing the efficient dimension of representations of input objects with specialized metrics. Since sketches do not require similarity estimation from a metric (see Sec. 1.3), question arises whether it is possible to directly create sketches with a constant distortion for such initial representations without intermediate embedding into L_s. As is shown in [75], this is impossible for initial representations from normed vector spaces (they should not necessarily be L_s) in the following distance threshold estimation problem: determine from sketches whether objects are similar or dissimilar. Of interest is the obtaining of similar results for a wider class of metrics (from nonnormalized spaces such as $\text{dist}_{\text{edit}}$).

8.3. Fast similarity search. Linear similarity search with the use of fast estimation of distance/similarity between a query object and all the objects of a base allows one to decrease the time of linear search based on initial similarity measures but does not rigorously guarantee the quality of the obtained results.

For similarity search, the exact estimate for distances in their entire range and between all objects is redundant. It suffices to correctly estimate the relationship between distances (large or small). Moreover, high precision is necessary only for small distances. This potentially allows one to use not only oblivious embeddings and sketches developed for fast and exact estimation of similarities and distances (for example, with satisfying versions of JL lemmas) but also other ones as well as those of smaller dimensions. A formalization of oblivious embeddings for searching for the approximate nearest neighbor and an example of such embeddings for the distance L_2 and data with a small intrinsic dimension are presented in [5].

As has been noted in Sec. 4, for L_1, there is no embeddings satisfying the JL lemma. However, for linear embeddings with the use of a random Cauchy i.i.d. matrix, there is a “one-sided” analogue of the JL lemma, which provides the search for approximate nearest neighbors for L_1 on the basis of such embeddings with $d << D$ [81]. Nonlinear embeddings into L_1 for small distances are given in [80] and in Sec. 4.1.

Fast estimation of distances also accelerates similarity search with the use of existing algorithms (index structures) operating on the basis of computation of distances [143–145]. Though similarity search is approximate in the general case as a result of inaccuracy of estimates, exact search results can be obtained for contractive estimates (see Sec. 6.1). Moreover, an acceleration of similarity search is possible owing to the use of algorithms and structures specialized for obtained real vectors (of small and moderate dimension) with their distance/similarity measures, for example, based on trees [145, 146] or locality-sensitive hashing (LSH) [9, 98, 147].

9. DISCUSSION

We now sum up advantages and drawbacks of the considered methods of formation of real-valued vector representations for estimating distance/similarity measures and compare them with methods that use learning.

9.1. Advantages and drawbacks of embeddings of vectors of Euclidean space with the help of random projection. Advantages of embeddings of vectors of a Euclidean space (see Secs. 2 and 3), i.e., real-valued vectors, for which the Euclidean distance, dot product, and angle are defined and can be estimated, by a random projection are as follows: a small distortion of estimates with a small dimension of final vectors; linearity of obtaining; allowing for all components of an input vector and suitability for any initial vectors (nonsparse and sparse, real-valued and binary, and with “heavy tails”); the possibility of streaming processing with a linear model; a developed apparatus of analysis in terms of average error and for the worst case (versions of the JL lemma).

In some cases, the obtained real-valued vectors (of small dimension) can be used directly in some index structures of fast similarity search, in linear and nonlinear vector methods of classification, approximation, and others and also for the subsequent quantization of components [1, 147, 148].

Drawbacks are as follows: the need for the formation of random matrices; complexity of multiplication by a matrix (but see the acceleration of projection in Sec. 3); inapplicability to streaming processing models with arbitrary weighting; impossibility of estimation of distances between a subset of components of initial vectors; nonsparseness (for any initial vectors including sparse ones).

9.2. Advantages and drawbacks of sketches obtained by stable random projections. Advantages of sketches for estimating L_s distances ($0< s \leq 2$) obtained by a stable random i.i.d. projection (see Sec. 4) are similar to those mentioned in Sec. 9.1. Note that, for a number of non-vector object representations, there is an embedding into L_1 and, therefore, for L_1, distances between initial representations can be estimated from sketches (see Sec. 6.2.).

In addition to the drawbacks considered in Sec. 9.1, the following drawbacks can be mentioned: the need for the formation of different random matrices for each value of s and complexity of generation of random numbers from stable distributions; nonlinearity of estimates; insufficient investigation of the acceleration of random projection (but see sparse random matrices in [149] from an s-Pareto distribution); impossibility of "automatic" application in a number of methods directly operating with vectors.

9.3. Advantages and drawbacks of sketches obtained by sampling. The advantages of sketches obtained by a uniform random sampling without replacement (see Secs. 5.1 and 5.2) are as follows: suitability of the same sketch for estimating any linear summary statistics; simplicity of obtaining a sketch; possibility of more exact estimates for sparse vectors; applicability to any streaming processing models including models with arbitrary weighting of components (initial vectors); possibility of operating with singled out subsets of components.

Drawbacks are as follows: low accuracy of estimates for nonsparse data and data with heavy tails; in the majority of cases, the complexity of analysis of the error of an estimate and lack of worst case guarantees.

There are problems with direct application of CRS sketches (see Sec. 5.2) in vector algorithms and LSH. Components of different sketches with the same number in a sketch do not correspond to one another, and, hence, for example, when linear models are learned, they should be unfolded into vectors of the initial dimension. In [62], problems with the construction of a kernel similarity matrix from them are also mentioned.

For vectors with heavy tails, weighted sampling methods allow one to increase the accuracy of estimates, but they operate with nonnegative input vectors, require the development of estimates for different similarities and distances, and estimation and computation of their errors are nontrivial. If heavy tails are absent, then the results of simple sampling can be better [84].

9.4. Learning-based methods for similarity estimation. The majority of methods of formation of vector representations for fast distance/similarity estimation that are considered in this survey do not take into account distinctive features of data of a concrete base. Adaptation to data opens possibilities of improvement in the results of applying fast distance/similarity estimation. For example, in similarity search, acceleration can be obtained owing to the formation of more compact representations, and also search quality can be increased owing to fine tuning to the base of representations and distance/similarity measures being used.

The dimensionality reduction of vector representations with the use of unsupervised and supervised learning is performed using linear and nonlinear methods [150, 151]. An example of a linear (contractive) transformation formed using unsupervised learning is the principal component analysis (PCA) method. Projection directions are determined by means of singular value decomposition (SVD) of a data matrix. With dimensionality reduction, PCA provides the smallest (for linear methods) mean-square error of estimates of Euclidean distances between vectors of a training set. However, the distance between a concrete pair of vectors can have an arbitrary distortion (worst-case guarantees are absent). Moreover, the dimension of the embedding providing a given distortion is not computed theoretically. Learning methods are also applied to the formation of compact binary vector representations reflecting the similarity of input objects [147, 148].

To improve the quality of similarity search, metric learning is used [152, 153]. Information on distinctive features of similar and dissimilar objects is specified by a teacher and is used for tuning parameters of distance/similarity measures. For example, the matrix of parameters **A** of the Makhalonobis distance (see Sec. 8.1) is adjusted based on a training set.

A common drawback of learning methods is their high computational complexity. For some methods, the formation of vector representations of new objects that are not used in learning is nontrivial (but see the Nystrom method, Sec. 6.1). Learning methods for dimensionality reduction do not always solve the problem of preservation of initial distances/similarities and, therefore, they are subject to large distortions without guarantees of preservation or reduction of distances. Moreover, adaptation to data presumes that the data of a training set and new data will have the same distribution, which does not always take place in practice.

The author is grateful to A. M. Sokolov for discussions.

REFERENCES

1. D. A. Rachkovskij, "Binary vectors for fast distance and similarity estimation," Cybernetics and Systems Analysis, **53**, No. 1 (2017) (to be printed).
2. M. Deza and E. Deza, Encyclopedia of Distances, Springer, Berlin-Heidelberg (2016).
3. P. Indyk and J. Matousek, "Low-distortion embeddings of finite metric spaces," in: Handbook of Discrete and Computational Geometry, Chapman & Hall/CRC, Boca Raton (FL) (2004), pp. 177–196.
4. J. Matousek, Lecture Notes on Metric Embeddings (2013).
5. P. Indyk and A. Naor, "Nearest-neighbor-preserving embeddings," ACM Trans. Algorithms, **3**, No. 3, Article No. 31 (2007).
6. T. Batu, F. Ergun, and C. Sahinalp, "Oblivious string embeddings and edit distance approximations," SODA'06, 792–801 (2006).
7. G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine, "Synopses for massive data: Samples, histograms, wavelets, sketches," Foundations and Trends® in Databases, **4**, Nos. 1–3, 1–294 (2012).
8. W. B. Johnson and J. Lindenstrauss, "Extensions of Lipshitz mapping into Hilbert space," Contemporary Mathematics, **26**, 189–206 (1984).
9. P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in: Proc. 30th ACM Symp. on Theory of Computing (1998), pp. 604–613.
10. D. Achlioptas, "Database-friendly random projections: Johnson–Lindenstrauss with binary coins," Journal of Computer and System Sciences, **66**, No. 4, 671–687 (2003).
11. J. Matousek, "On variants of the Johnson Lindenstrauss lemma," Random Structures and Algorithms, **33**, No. 2, 142–156 (2008).
12. T. S. Jayram and D. P. Woodruff, "Optimal bounds for Johnson–Lindenstrauss transforms and streaming problems with subconstant error," ACM Trans. on Algorithms, **9**, No. 3, Article 26 (2013).
13. D. M. Kane, R. Meka, and J. Nelson, "Almost optimal explicit Johnson–Lindenstrauss families," in: Proc. RANDOM'11 (2011), pp. 628–639.
14. K. G. Larsen and J. Nelson, "The Johnson–Lindenstrauss lemma is optimal for linear dimensionality reduction," in: Proc. ICALP'16 (2016).
15. K. G. Larsen and J. Nelson, Optimality of the Johnson–Lindenstrauss Lemma, arXiv:1609.02094.
16. R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," in: Compressed Sensing, Theory and Applications (2012), pp. 210–268.
17. V. Buldygin and K. Moskvichova, "The sub-Gaussian norm of a binary random variable," Theory of Probability and Mathematical Statistics, **86**, 33–49 (2013).
18. S. Dirksen, "Dimensionality reduction with sub-Gaussian matrices: A unified theory," Foundations of Computational Mathematics, 1–30 (2015).
19. R. G. Baraniuk, M. Davenport, R. A. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," Constr. Approx., **28**, No. 3, 253–263 (2008).
20. J. Nelson, E. Price, and M. Wootters, "New constructions of RIP matrices with fast multiplication and fewer rows," in: Proc. SODA'14 (2014), pp. 1515–1528.

21. F. Krahmer and R. Ward, "New and improved Johnson–Lindenstrauss embeddings via the Restricted Isometry Property," SIAM J. Math. Anal., **43**, No. 3, 1269–1281 (2011).
22. N. Ailon and H. Rauhut, "Fast and RIP-optimal transforms," Discrete and Computational Geometry, **52**, No. 4, 780–798 (2014).
23. I. Haviv and O. Regev, "The restricted isometry property of subsampled Fourier matrices," in: Proc. SODA'16, 288–297 (2016).
24. J. Bourgain, S. Dirksen, and J. Nelson, "Toward a unified theory of sparse dimensionality reduction in Euclidean space," Geometric and Functional Analysis, **25**, No. 4, 1009–1088 (2015).
25. Y. Gordon, "On Milman's inequality and random subspaces which escape through a mesh in R^n," Geometric Aspects of Functional Analysis, 84–106 (1988).
26. G. Schechtman, "Two observations regarding embedding subsets of Euclidean spaces in normed spaces," Adv. Math., **200**, No. 1, 125–135 (2006).
27. S. Oymak, B. Recht, and M. Soltanolkotabi, Isometric Sketching of Any Set via the Restricted Isometry Property, arXiv:1506.03521 (2015).
28. B. Klartag and S. Mendelson, "Empirical processes and random projections," Journal of Functional Analysis, **225**, No. 1, 229–245 (2005).
29. Z. Karnin, Y. Rabani, and A. Shpilka, "Explicit dimension reduction and its applications," SIAM J. Comput., **41**, No. 1, 219–249 (2012).
30. D. M. Kane and J. Nelson, "Sparser Johnson-Lindenstrauss transforms," Journal of the ACM, **61**, No. 1, 4:1–4:23 (2014).
31. L. Engebretsen, P. Indyk, and R. O'Donnell, "Derandomized dimensionality reduction with applications," in: Proc. SODA'02, 705–712 (2002).
32. D. Sivakumar, "Algorithmic derandomization via complexity theory," in: Proc. 34th Annual ACM Symposium on Theory of Computing, Montreal, QC (2002); ACM, New York (2002), pp. 619–626.
33. P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in: Proc. KDD'06 (2006), pp. 287–296.
34. D. A. Rachkovskij, "Vector data transformation using random binary matrices," Cybernetics and Systems Analysis, **50**, No. 6, 960–968 (2014).
35. S. S. Vempala, The Random Projection Method, American Math. Soc. (2004).
36. R. I. Arriaga and S. Vempala, "An algorithmic theory of learning: Robust concepts and random projection," Machine Learning, **63**, No. 2, 161–182 (2006).
37. A. Kabán, "Improved bounds on the dot product under random projection and random sign projection," in: Proc. KDD'15 (2015), pp. 487–496.
38. X. Yi, C. Caramanis, and E. Price, Binary Embedding: Fundamental Limits and Fast Algorithm, arXiv:1502. 05746 (2015).
39. A. Magen, "Dimensionality reductions in $\ell2$ that preserve volumes and distance to affine spaces," Discrete Comput. Geom., **38**, No. 1, 139–153 (2007).
40. Y. Plan and R. Vershynin, "One-bit compressed sensing by linear programming," Communications on Pure and Applied Mathematics, **66**, No. 8, 1275–1297 (2013).
41. N. Ailon and B. Chazelle, "The Fast Johnson–Lindenstrauss Transform and approximate nearest neighbors," SIAM J. Comput., **39**, No. 1, 302–322 (2009).
42. Y. Plan and R. Vershynin, "Dimension reduction by random hyperplane tessellations," Discrete and Computational Geometry, **51**, No. 2, 438–461 (2014).
43. E. Liberty and S. W. Zucker, "The Mailman algorithm: A note on matrix-vector multiplication," Inf. Process. Lett., **109**, No. 3, 179–182 (2009).
44. D. Rachkovskij and S. Slipchenko, "Similarity-based retrieval with structure-sensitive sparse binary distributed representations," Computational Intelligence, **28**, No. 1, 106–129 (2012).
45. P. Kanerva, J. Kristoferson, and A. Holst, "Random indexing of text samples for latent semantic analysis," in: Proc. 22nd Annual Conference of the Cognitive Science Society (2000), p. 1036.

46. I. S. Misuno, D. A. Rachkovskij, and S. V. Slipchenko, “Vector and distributed representations reflecting semantic relatedness of words,” Mathematical Machines and Systems, No. 3, 50–67 (2005).

47. D. A. Rachkovskij, I. S. Misuno, and S. V. Slipchenko, “Randomized projective methods for construction of binary sparse vector representations,” Cybernetics and Systems Analysis, **48**, No. 1, 146–156 (2012).

48. D. A. Rachkovskij, “Formation of similarity-reflecting binary vectors with random binary projections,” Cybernetics and Systems Analysis, **51**, No. 2, 313–323 (2015).

49. N. Ailon and E. Liberty, “Fast dimension reduction using Rademacher series on dual BCH codes,” Discrete and Computational Geometry, **42**, No. 4, 615–630 (2009).

50. E. Liberty, N. Ailon, and A. Singer, “Dense fast random projections and lean Walsh transforms,” Discrete and Computational Geometry, **45**, No. 1, 34–44 (2011).

51. N. Ailon and E. Liberty, “An almost optimal unrestricted fast Johnson–Lindenstrauss transform,” ACM Transactions on Algorithms, **9**, No. 3. Article No 21 (2013).

52. H. Rauhut, J. Romberg, and J. Tropp, “Restricted isometries for partial random circulant matrices,” Applied and Computational Harmonic Analysis, **32**, No. 2, 242–254 (2012).

53. A. Hinrichs and J. Vybiral, “Johnson-Lindenstrauss lemma for circulant matrices,” Random Structures & Algorithms, **39**, No. 3, 391–398 (2011).

54. J. Vybiral, “A variant of the Johnson–Lindenstrauss lemma for circulant matrices,” Journal of Functional Analysis, **260**, No. 4, 1096–1105 (2011).

55. F. Krahmer, S. Mendelson, and H. Rauhut, “Suprema of chaos processes and the restricted isometry property,” Comm. Pure Appl. Math., **67**, No. 11, 1877–1904 (2014).

56. H. Zhang and L. Cheng, “New bounds for circulant Johnson–Lindenstrauss embeddings,” Communications in Mathematical Sciences, **12**, No. 4, 695–705 (2014).

57. Z. Yang, M. Moczulski, M. Denil, N. de Freitas, A. Smola, L. Song, and Z. Wang, “Deep fried convents,” in: Proc. ICCV’15 (2015), pp. 1476–1483.

58. Y. Cheng, F. X. Yu, R. S. Feris, S. Kumar, A. Choudhary, and S.- F. Chang, “An exploration of parameter redundancy in deep networks with circulant projections,” in: Proc. ICCV’15 (2015), pp. 2857–2865.

59. V. Sindhwani, T. Sainath, and S. Kumar, “Structured transforms for small-footprint deep learning,” in: Proc. NIPS’15 (2015), pp. 3070–3078.

60. M. Moczulski, M. Denil, J. Appleyard, N. de Freitas, “ACDC: A structured efficient linear layer,” in: ICLR’16, arXiv:1511.05946 (2016).

61. K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, “Feature hashing for large scale multitask learning,” in: Proc. ICML’09 (2009), pp. 1113–1120.

62. P. Li, A. Shrivastava, J. L. Moore, and A. C. König, “Hashing algorithms for large-scale learning,” in: Proc. NIPS’11 (2011), pp. 2672–2680.

63. A. Dasgupta, R. Kumar, and T. Sarlos, “A sparse Johnson–Lindenstrauss transform,” in: Proc. STOC’10 (2010), pp. 341–350.

64. D. M. Kane and J. Nelson, “A derandomized sparse Johnson–Lindenstrauss transform,” Electronic Colloquium on Computational Complexity, **17**, Article 98 (2010).

65. V. Braverman, R. Ostrovsky, and Y. Rabani, “Rademacher chaos, random Eulerian graphs and the sparse Johnson–Lindenstrauss transform,” arXiv:1011.2590 (2010).

66. J. Nelson and H. L. Nguyen, “Sparsity lower bounds for dimensionality reducing maps,” in: Proc. STOC’13 (2013), pp. 101–110.

67. S. Ventkatasubramanian and Q. Wang, “The Johnson-Lindenstauss transform: An empirical study,” in: Proc. ALENEX’11 (2011), pp. 164–173.

68. J. R. Lee and A. Naor, “Embedding the diamond graph in Lp and dimension reduction in L1,” Geometric and Functional Analysis, **14**, No. 4, 745–747 (2004).

69. B. Brinkman and M. Charikar, “On the impossibility of dimension reduction in L1,” Journal of the ACM, **52**, No. 5, 766–788 (2005).

70. J. Lee, M. Mendel, and A. Naor, “Metric structures in L1: Dimension, snowflakes, and average distortion,” European Journal of Combinatorics, **26**, No. 8, 1180–1190 (2005).
71. A. Andoni, M. Charikar, O. Neiman, and H. L. Nguyen, “Near linear lower bounds for dimension reduction in L1,” in: Proc. FOCS’11 (2011), pp. 315–323.
72. I. Newman and Y. Rabinovich, Finite Volume Spaces and Sparsification, arXiv:1002.3541 (2010).
73. T. Figiel, J. Lindenstrauss, and V. D. Milman, “The dimension of almost spherical sections of convex bodies,” Acta Math., **139**, No. 1, 53–94 (1977).
74. W. B. Johnson and G. Schechtman, “Embedding l_p^m into l_1^n,” Acta Math., **149**, No 1, 71–85 (1982).
75. A. Andoni, R. Krauthgamer, and I. P. Razenshteyn, “Sketching and embedding are equivalent for norms,” in: Proc. STOC’15 (2015), pp. 479–488.
76. Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar, “An information statistics approach to data stream and communication complexity,” J. Comput. Syst. Sci., **68**, No. 4, 702–732 (2004).
77. R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, M. J. Strauss, “Combining geometry and combinatorics: A unified approach to sparse signal recovery,” in: AAC on CCC’08 (2008), pp. 798–805.
78. F. Krahmer and R. Ward, “A unified framework for linear dimensionality reduction in L1,” Results in Mathematics, **70**, No. 1, 209–231 (2016).
79. Z. Allen-Zhu, R. Gelashvili, and I. Razenshteyn, “Restricted isometry property for general p-norms,” in: Proc. SoCG’15 (2015), pp. 451–460.
80. Y. Bartal and L.-A. Gottlieb, “Dimension reduction techniques for ℓp ($1 \le p \le 2$), with applications,” in: Proc. SoCG’16 (2016), pp. 16:1–16:15.
81. P. Indyk, “Stable distributions, pseudorandom generators, embeddings, and data stream computation,” Journal of the ACM, **53**, No. 3, 307–323 (2006).
82. P. Li, “Estimators and tail bounds for dimension reduction in $\ell\alpha$ ($0< \alpha \le 2$) using stable random projections,” in: Proc. SODA’08 (2008), pp. 10–19.
83. P. Li, “Computationally efficient estimators for dimension reductions using stable random projections,” in: Proc. ICDM’08 (2008), pp. 403–412.
84. N. Duffield, C. Lund, and M. Thorup, “Priority sampling for estimating arbitrary subset sums,” J. Assoc. Comput. Mach., **54**, No. 6, Article No 32 (2007).
85. E. Cohen, “Distance queries from sampled data: Accurate and efficient,” in: Proc. KDD’14 (2014), pp. 681–690.
86. P. Li, K. W. Church, and T. J. Hastie, “One sketch for all: Theory and applications of conditional random sampling,” in: Proc. NIPS’08 (2008), pp. 953–960.
87. M. Thorup, “Bottom-k and priority sampling, set similarity and subset sums with minimal independence,” in: Proc. STOC’13 (2013), pp. 371–378.
88. M. Charikar, “Similarity estimation techniques from rounding algorithms,” in: Proc. STOC’02 (2002), pp. 380–388.
89. S. Ioffe, “Improved consistent sampling, weighted minhash and L1 sketching,” in: Proc. ICDM’10 (2010), pp. 246–255.
90. P. Li, Generalized Min-Max Kernel and Generalized Consistent Weighted Sampling, arXiv:1605.05721 (2016).
91. E. Cohen, “Estimation for monotone sampling: Competitiveness and customization,” in: Proc. PODC’14 (2014), pp. 124–133.
92. C. K. I. Williams and M. Seeger, “Using the Nystrom method to speed up kernel machines,” in: Proc. NIPS’00 (2000), pp. 682–688.
93. G. R. Hjaltason and H. Samet, “Properties of embedding methods for similarity searching in metric spaces,” IEEE Trans. PAMI, **25**, No. 5, 530–549 (2003).
94. J. C. Platt, “FastMap, MetricMap, and Landmark MDS are all Nystrom algorithms,” in: Proc. AISTATS’05 (2005), pp. 261–268.
95. E. Pekalska and R. P. W. Duin, The Dissimilarity Representation for Pattern Recognition: Foundations and Applications, World Scientific, Singapore (2005).

96. E. Chavez, M. Graff, G. Navarro, and E. S. Tellez, "Near neighbor searching with K nearest references," Information Systems, **51(C)**, 43–61 (2015).

97. K. Riesen, M. Neuhaus, and H. Bunke, "Graph embedding in vector spaces by means of prototype selection," in: Proc. GbRPR'07 (2007), pp. 383–393.

98. A. Andoni, Nearest Neighbor Search: The Old, the New, and the Impossible, PhD Thesis, Massachusetts Institute of Technology (2009).

99. V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Soviet Physics — Doklady, **10**, No. 8, 707–710 (1966).

100. G. Navarro, "A guided tour to approximate string matching," ACM CSUR, **33**, No. 1, 31–88 (2001).

101. A. Backurs and P. Indyk, "Edit distance cannot be computed in strongly subquadratic time (unless SETH is false)," in: Proc. STOC'15 (2015), pp. 51–58.

102. K. Bringmann and M. Kunnemann, "Quadratic conditional lower bounds for string problems and dynamic time warping," in: Proc. FOCS'15 (2015), pp. 79–97.

103. G. Cormode and S. Muthukrishnan, "The string edit distance matching problem with moves," ACM Trans. Algorithms, **3**, No. 1, 2:1–2:19 (2007).

104. A. M. Sokolov, "Vector representations for efficient comparison and search for similar strings," Cybernetics and System Analysis, **43**, No. 4, 484–498 (2007).

105. R. Ostrovsky and Y. Rabani, "Low distortion embeddings for edit distance," Journal of the ACM, **54**, No. 5, 23–36 (2007).

106. A. Andoni and K. Onak, "Approximating edit distance in near-linear time," SIAM Journal on Computing, **41**, No. 6, 1635–1648 (2012).

107. A. Andoni, R. Krauthgamer, and K. Onak, "Polylogarithmic approximation for edit distance and the asymmetric query complexity," in: Proc. FOCS'10 (2010), pp. 377–386.

108. R. Krauthgamer and Y. Rabani, "Improved lower bounds for embeddings into L1," in: Proc. SODA'06 (2006), pp. 1010–1017.

109. A. Andoni, A. Goldberger, A. McGregor, and E. Porat, "Homomorphic fingerprints under misalignments: Sketching edit and shift distances," in: Proc. STOC'13 (2013), pp. 931–940.

110. B. Scholkopf and A. J. Smola, Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge (2001).

111. S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," Journal of Machine Learning Research, **11**, 1201–1242 (2010).

112. D. Conte, J. Y. Ramel, N. Sidere, M. M. Luqman, B. Gauzere, J. Gibert, L. Brun, and M. Vento, "A comparison of explicit and implicit graph embedding methods for pattern recognition," LNCS, **7877**, 81–90 (2013).

113. A. Feragen, N. Kasenburg, J. Petersen, M. de Bruijne, and K. M. Borgwardt, "Scalable kernels for graphs with continuous attributes," in: Proc. NIPS'13 (2013), pp. 216–224.

114. P. Foggia, G. Percannella, and M. Vento, "Graph matching and learning in pattern recognition in the last 10 years," Int. J. Pattern Recog. Artif. Intell., **28**, No. 1, 1–40 (2014).

115. N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," SIAM Review, **53**, No. 2, 217–288 (2011).

116. A. Gittens and M. W. Mahoney, "Revisiting the Nystrom method for improved large-scale machine learning," in: Proc. ICML'13 (2013), pp. 567–575.

117. M. B. Cohen, Y. T. Lee, C. Musco, C. Musco, R. Peng, and A. Sidford, "Uniform sampling for matrix approximation," in: Proc. ITCS'15 (2015), pp. 181–190.

118. S. Wang, Luo Luo, Zhihua Zhang, "SPSD matrix approximation vis column selection: Theories, algorithms, and extensions," Journal of Machine Learning Research, **17**, 1–49 (2016).

119. N. Shervashidze, S. V. N. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," JMLR: W&CP, **5**, 488–495 (2009).

120. J. Gibert, E. Valveny, and H. Bunke, "Embedding of graphs with discrete attributes via label frequencies," Int. J. Patt. Recogn. Artif. Intell., **27**, No. 3, 1–27 (2013).

121. N. Kriege, M. Neumann, K. Kersting, and P. Mutzel, "Explicit versus implicit graph feature maps: A computational phase transition for walk kernels," in: Proc. ICDM'14 (2014), pp. 881–886.

122. A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in: Proc. NIPS'07 (2007), pp. 1177–1184.

123. J. A. Tropp, "An introduction to matrix concentration inequalities," Foundations and Trends® in Machine Learning, **8**, Nos. 1–2, 1–230 (2015).

124. T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou, "Nystrom method vs random Fourier features: A theoretical and empirical comparison," in: Proc. NIPS'12 (2012), pp. 485–493.

125. D. J. Sutherland and J. Schneider, "On the error of random Fourier features," in: Proc. UAI (2015), pp. 862–871.

126. B. K. Sriperumbudur and Z. Szabo, "Optimal rates for random Fourier features," in: Proc. NIPS'15 (2015), pp. 1144–1152.

127. D. Chen and J. M. Phillips, Relative Error Embeddings of the Gaussian Kernel Distance, arXiv:1602.05350.

128. J. Yang, V. Sindhwani, H. Avron, and M. W. Mahoney, "Quasi-Monte Carlo feature maps for shift-invariant kernels," in: Proc. ICML'14 (2014), pp. 485–493.

129. Q. Le, T. Sarlos, and A. J. Smola, "Fastfood — Computing Hilbert space expansions in loglinear time," JMLR: W&CP, **28**, No. 3, 244–252 (2013).

130. C. Feng, Q. Hu, and S. Liao, "Random feature mapping with signed circulant matrix projection," in: Proc. IJCAI'15 (2015), pp. 3490–3496.

131. F. X. Yu, S. Kumar, H. Rowley, and S.-F. Chang, Compact Nonlinear Maps and Circulant Extensions, arXiv:1503.03893 (2015).

132. K. Choromanski and V. Sindhwani, "Recycling randomness with structure for sublinear time kernel expansions," in: Proc. ICML (2016), pp. 2502–2510.

133. A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," IEEE Trans. PAMI, **34**, No. 3, 480–492 (2012).

134. J. Yang, V. Sindhwani, Q. Fan, H. Avron, and M. W. Mahoney, "Random Laplace feature maps for semigroup kernels on histograms," in: Proc. CVPR'14 (2014), pp. 971–978.

135. P. Kar and H. Karnick, "Random feature maps for dot product kernels," in: Proc. ICAIS'12 (2012), pp. 583–591.

136. N. Pham and R. Pagh, "Fast and scalable polynomial kernels via explicit feature maps," in: Proc. KDD'13 (2013), pp. 239–247.

137. R. Hamid, Y. Xiao, A. Gittens, and D. DeCoste, "Compact random feature maps," in: Proc. ICML'14 (2014), pp. 19–27.

138. J. Pennington, F. X. Yu, and S. Kumar, "Spherical random features for polynomial kernels," in: Proc. NIPS'15 (2015), pp. 1846–1854.

139. A. Bhattacharya, P. Kar, and M. Pal, "On low distortion embeddings of statistical distance measures into low dimensional spaces." In: Proc. DEXA'09 (2009), pp. 164–172.

140. R. J. Kyng, J. M. Phillips, and S. Venkatasubramanian, "Johnson–Lindenstrauss dimensionality reduction on the simplex," in: Proc. FWCG'10 (2010).

141. A. Abdullah, A. McGregor, R. Kumar, S. Vassilvitskii, and S. Venkatasubramanian, "Sketching, embedding, and dimensionality reduction in information spaces," JMLR: W&CP, **41**, 948–956 (2016).

142. S. Guha, P. Indyk, and A. McGregor, "Sketching information divergences," in: COLT (2007), pp. 424–438.

143. E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín, "Searching in metric spaces," ACM Computing Surveys, **33**, No. 3, 273–321 (2001).

144. P. Zezula, G. Amato, V. Dohnal, and M. Batko, Similarity Search: The Metric Space Approach," Springer, New York (2006).

145. H. Samet, Foundations of Multidimensional and Metric Data Structures, Morgan Kaufmann, San Francisco (2006).

146. M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," IEEE Trans. on PAMI, **36**, No. 11, 2227–2240 (2014).

147. J. Wang, H. T. Shen, J. Song, and J. Ji, Hashing for Similarity Search: A Survey, arXiv:1408.2927 (2014).

148. J. Wang, W. Liu, S. Kumar, and S.-F. Chang, “Learning to hash for indexing big data: A survey,” in: Proc. IEEE, **104**, No. 1, 34–57 (2016).

149. P. Li, “Very sparse stable random projections for dimension reduction in $l\alpha$ $(0 < \alpha \leq 2)$ norm,” in: Proc. SIGKDD’07 (2007), pp. 440–449.

150. J. Cunningham and Z. Ghahramani, “Linear dimensionality reduction: Survey, insights, and generalizations,” Journal of Machine Learning Research, **16**, 2859–2900 (2015).

151. L. J. P. Van der Maaten, E. O. Postma, and H. J. Van den Herik, “Dimensionality reduction: A comparative review,” Tilburg University Technical Report, TiCC-TR 2009-005 (2009).

152. B. Kulis, “Metric learning: A survey,” Foundations and Trends® in Machine Learning, **5**, No. 4, 287–364 (2012).

153. A. Bellet, A. Habrard, and M. Sebban, A Survey on Metric Learning for Feature Vectors and Structured Data, arXiv:1306.6709 (2014).