НОВІ ЗАСОБИ
КІБЕРНЕТИКИ, ІНФОРМАТИКИ,
ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
ТА СИСТЕМНОГО АНАЛІЗУ

Д.А. РАЧКОВСКИЙ

УДК 004.22 + 004.93'11

ВЕЩЕСТВЕННЫЕ ВЛОЖЕНИЯ И СКЕТЧИ ДЛЯ БЫСТРОЙ ОЦЕНКИ РАССТОЯНИЙ И СХОДСТВ

Аннотация. Рассмотрены методы и алгоритмы быстрой оценки мер расстояния/сходства данных по формируемым вещественным векторам малой размерности. Приведены методы без обучения, использующие главным образом случайное проецирование и сэмплирование. Исходные данные являются в основном векторами большой размерности с различными расстояниями (евклидовым, манхэттеновым, статистическими и др.) и сходствами (скалярным произведением и др.). Обсуждаются и векторные представления невекторных данных. Получаемые векторы можно также применять в алгоритмах поиска по сходству, машинного обучения и др.

Ключевые слова: расстояние, сходство, вложения, скетчи, снижение размерности, случайное проецирование, сэмплирование, лемма Джонсона–Линденштраусса, ядерное сходство, поиск по сходству.

1. ОСНОВНЫЕ ПОНЯТИЯ

Функции (меры) расстояния и сходства широко используются как в поиске по сходству, так и во многих приложениях анализа данных, машинного обучения, статистики (кластерный анализ, классификация и аппроксимация методом ближайшего соседа, многомерное шкалирование и др.). Для сложно вычислимых расстояний/сходств актуальна их быстрая оценка или вычисление границ значений. Для получения такой оценки часто используют преобразование исходных представлений данных (объектов) различного типа (векторных и невекторных) с разными мерами расстояния/сходства в представления (обычно векторные, малой размерности), которые позволяют вычислительно легко оценить сходство исходных данных. Сложность оценки расстояний/сходств между векторами (например, евклидова расстояния или скалярного произведения и др.) линейна от их размерности, поэтому при небольшой размерности сложность невелика. Для векторных представлений также существует множество методов поиска по сходству, статистического распознавания образов, классификации, кластеризации, аппроксимации, отбора информативных признаков и др.

В настоящей статье дан обзор подходов, методов и алгоритмов быстрой оценки расстояний/сходств исходных представлений данных по вещественным векторным представлениям. (Бинарные и целочисленные векторные представления для быстрой оценки расстояний/сходств рассмотрены в [1].) Представлены в основном методы без адаптации к данным (но см. подразд. 9.4). Большинство рассмотренных методов и алгоритмов практически реализуемы, хотя в некоторых случаях даны только теоретические границы на значения параметров. (Ввиду ограниченного объема статьи цитируются в основном публикации, содержащие ссылки на предыдущие работы.)

1.1. Расстояния и сходства. Для каждого типа представления данных (объектов) существуют различные меры расстояния/сходства. Число типов представлений невелико. Наиболее распространены векторные представления, а также множества, последовательности, деревья и графы.

Например, если объекты представлены в виде совокупности числовых признаков (как вещественные векторы $\mathbf{x}$, $\mathbf{y}$ размерности D), сходством может являться скалярное произведение $\mathrm{sim}_{\,\mathrm{dot}}(\mathbf{x}, \mathbf{y}) \equiv \langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{D} x_i y_i$. Бóльшие значения сходств соответствуют более схожим объектам. Для оценки сходства объектов используют также расстояние, т.е. «несходство» (например, угол или евклидово расстояние между векторами). Большим значениям сходства соответствуют малые значения расстояния. Многие расстояния являются метриками, т.е. подчиняются таким метрическим аксиомам, как неравенство треугольника и др. Свойства мер сходства можно использовать для ускорения поиска по сходству, например, для метрик используют неравенство треугольника.

Для векторов широко применяются расстояния Минковского L_s различного порядка s: $|| \mathbf{x} - \mathbf{y} ||_s = \left(\sum_{i=1}^{D} | x_i - y_i |^s \right)^{1/s}$. Наиболее распространены метрические расстояния L_2 (евклидово $|| \mathbf{x} - \mathbf{y} ||_2$), L_1 (манхэттеново $|| \mathbf{x} - \mathbf{y} ||_1$), L_∞ (расстояние Чебышева или максимума $|| \mathbf{x} - \mathbf{y} ||_\infty$). При $0 < s < 1$ получаются дробные расстояния, которые не являются метриками. Сложность вычисления расстояния Минковского между двумя векторами составляет $O(D)$. Векторное пространство с расстоянием L_s также обозначим L_s (или L_s^D при его размерности D). Многие типы расстояний/сходств для различных представлений объектов рассмотрены в [2]. Определения расстояний в настоящем обзоре приводятся по мере их использования.

1.2. Вложения. Преобразование $\mathrm{f}(x)$ множества объектов исходного пространства в целевое (обычно более «легкое»), сохраняющее исходные расстояния, называют вложением. Хотя целевыми могут являться пространства различного типа (например, метрики деревьев [3] и др.), в основном применяют вложения в нормированные (векторные) пространства L_s.

Качество вложений оценивают [3, 4] искажением расстояний. Часто для малых $\varepsilon > 0$ используют мультипликативное искажение — минимальное ε, при котором

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

Назовем его $1 \pm \varepsilon$-искажением, или искажением $1 + \varepsilon$. Для больших искажений $A > 1$ выражение для мультипликативного A-искажения имеет вид

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

Аддитивное искажение $\pm \varepsilon_a$, $\varepsilon_a > 0$, определим как

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

Искажения можно также определить не для расстояний dist, а для сходств sim.

При $\varepsilon = 0$, $\varepsilon_a = 0$, $A = 1$ в выражениях (1)–(3) расстояния/сходства сохраняются точно, т.е. имеет место изометрия, например вложение Фреше N объектов метрического пространства в L_∞, всего пространства L_1 в L_∞ ([3] и подразд. 6.1), ядерных сходств $\kappa(x, y)$ в гильбертово пространство H (разд. 7).

К недостаткам изометрических вложений относят следующие: расстояния сохраняются обычно только для заданного множества объектов, размерность вложений велика, и таких вложений известно мало. Поэтому востребованы при-

ближенные вложения в векторные пространства (малой размерности), позволяющие быстро оценить исходные расстояния с малым искажением [3, 4].

Для таких задач, как поиск по сходству, объекты-запросы часто заранее неизвестны, а состав объектов базы, где выполняется поиск, может изменяться. Поэтому требуются забывчивые (oblivious) методы формирования представлений объектов, которые можно применять к новым объектам [5] (не изменяя существующих представлений). В [6] забывчивые вложения объектов не зависят от других объектов (более сильное определение забывчивости). В настоящем обзоре большинство преобразований забывчивые согласно [6].

Поскольку вложение любых объектов с малым искажением является трудной задачей, забывчивые вложения обычно рандомизированные (выполняются с использованием псевдослучайных чисел и гарантируют искажение лишь с некоторой вероятностью). Например, рандомизированным является снижение размерности (разновидность вложения, где вид функции расстояния сохраняется, а размерность представлений уменьшается) векторов для евклидова расстояния по лемме JL с помощью случайных проекций (разд. 2).

1.3. Скетчи. Компактные представления исходных объектов, которые применяются для оценки некоторых их характеристик, называются скетчами [7]. Как и вложения, скетчи, используемые для оценки расстояний/сходств, обычно являются векторами.

Для оценки расстояний исходных объектов по скетчам можно использовать не расстояние, как во вложениях, а другие характеристики (например, медианные оценки и т.п., разд. 4). Аналитическая зависимость исходного расстояния/сходства от некоторой величины, определяемой по скетчам, бывает сложной или неизвестной, и для получения оценок можно использовать табуляцию.

Часто скетчами называют компактные представления, применяемые для потоковой обработки [7] (когда представления объектов задают последовательностью компонентов или их приращений). Скетчами также называют векторы с бинарными или целочисленными (дискретными) компонентами [1]. Такие векторы легко обрабатываются и обычно занимают меньше памяти, чем исходные представления объектов. В настоящей статье скетчами будем называть векторные представления исходных объектов для оценки их расстояний/сходств (включая результаты вложений).

Время оценки исходного расстояния/сходства по скетчам размерности d обычно составляет $O(d)$ (линейная временна́я сложность алгоритма). Поэтому при $d < D$ исходных векторов получаем ускорение оценок. Для исходных представлений с худшей, чем линейная, сложностью вычисления расстояния/сходства ее снижение до линейной для скетчей также является источником ускорения. Отметим, что бывают полезны вложения и скетчи без снижения размерности (числа элементов) представлений, например, если для них существуют эффективные алгоритмы поиска по сходству, оценки мер расстояний/сходств либо другие алгоритмы или методы, требующие использования представлений полученного типа [4].

1.4. Структура обзора. В разд. 2 рассматриваются снижение размерности векторов евклидова пространства случайным проецированием и искажения оценки евклидова расстояния (а также скалярного произведения, угла) исходных векторов по полученным скетчам. В разд. 3 обсуждается ускорение случайного проецирования. В разд. 4 приводятся вложения и скетчи для оценки неевклидовых расстояний Минковского.

В разд. 5 рассматриваются скетчи сэмплированием (отбором подмножества компонентов исходных представлений), в разд. 6 — вложения для оценки расстояний между невекторными данными, а в разд. 7 — аппроксимация ядерных сходств.

В разд. 8 описываются другие направления исследований, включая вложения статистических расстояний, эквивалентность скетчей и вложений, приближенный поиск по сходству. В разд. 9 приводятся преимущества и недостатки рассмотренных вещественных вложений и скетчей, дано сравнение с методами с обучением.

2. СНИЖЕНИЕ РАЗМЕРНОСТИ ВЕКТОРОВ ЕВКЛИДОВОГО ПРОСТРАНСТВА СЛУЧАЙНЫМ ПРОЕЦИРОВАНИЕМ

2.1. Леммы Джонсона–Линденштраусса. Возможность оценки евклидова расстояния L_2 исходных векторов с малым искажением по евклидову расстоянию между их вложениями малой размерности (т.е. снижение размерности) дает лемма JL (Johnson–Lindenstrauss).

Лемма JL [8, 4]. Для множества $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ любых N векторов в вещественном пространстве $\mathbb{R}^D$ (независимо от размерности D) существует вложение $\mathrm{f}: \mathbb{R}^D \to \mathbb{R}^d$, $d = O(\log N / \varepsilon^2)$ с искажением каждого из $N(N-1)/2$ расстояний L_2 между этими векторами не более $1 \pm \varepsilon$ (т.е. для них выполняется (1) для евклидовых dist_1 и dist_2).

Доказательства леммы JL [9–11] используют линейные рандомизированные вложения со свойствами, которые определяются вариантами так называемой леммы JL для распределений (DJL), также известной [4] как лемма случайных проекций (random projection). Эти DJL-леммы утверждают, что существуют классы распределений матриц размера $d \times D$ такие, что матрица $\mathbf{R}$, случайно выбранная из распределения, для любого вектора $\mathbf{z} \in \mathbb{R}^D$ с вероятностью Pr (по реализациям $\mathbf{R}$) обеспечивает мультипликативное $1 \pm \varepsilon$ искажение евклидовой нормы $\mathbf{z}$: $\Pr\{(1-\varepsilon)||\mathbf{z}||_2 \le ||\mathbf{Rz}||_2 \le (1+\varepsilon)||\mathbf{z}||_2\} \ge 1-\delta$ для любых $0 < \varepsilon,\ \delta < 1/2$ при $d = O(\min\{D, \varepsilon^{-2} \log(1/\delta)\})$. Часто леммы (D)JL формулируют в терминах квадратов норм и расстояний.

Доказательство лемм DJL основано на концентрации (для конкретных классов $\mathbf{R}$) $||\mathbf{Rz}||_2$ вокруг $||\mathbf{z}||_2$ (или $||\mathbf{Rz}||_2^2$ вокруг $||\mathbf{z}||_2^2$). Для леммы DJL (с линейным преобразованием $\mathbf{Rz}$) оптимальное значение $d = \Theta(\varepsilon^{-2} \log(1/\delta))$ [12, 13].

Из леммы DJL, выбирая $\delta < 2/N/(N-1)$ и полагая $\mathbf{z} = \mathbf{x} - \mathbf{y}$ для всех $N(N-1)/2$ пар векторов, получают выполнение леммы JL с большой вероятностью применением неравенства Буля (union bound). Отметим, что величина $1 \pm \varepsilon$ в лемме DJL не зависит от $||\mathbf{z}||_2$, а в лемме JL — от $||\mathbf{x} - \mathbf{y}||_2$. Поэтому в доказательствах можно считать норму этих векторов единичной.

Случайное проецирование обеспечивает забывчивость преобразования. Размерность $d = O(\varepsilon^{-2} \log N)$ в лемме JL оптимальна для линейного преобразования $\mathbf{Rz}$ [14] и в $\log(1/\varepsilon)$ раз больше оптимальной в общем случае [15].

Будем называть JLT классы матриц линейного преобразования, для которых выполняется лемма JL. Большим (но не единственным) классом JLT являются матрицы с элементами i.i.d. (независимо и одинаково распределенными) случайными величинами (с.в.) из субгауссового распределения [11, 16–18]. Центрированная с.в. x субгауссова, если $\exists\, c > 0\ \forall \lambda > 0\ \Pr\{|x| > \lambda\} \le 2\exp(-c\lambda^2)$. Например, субгауссовыми JLT являются матрицы с i.i.d. элементами из гауссова распределения $\mathrm{Norm}(0, 1)$, с бинарными элементами из $\{-1, +1\}$ с вероятностью 1/2 (распределение Радемахера), с тернарными элементами из $\{-1/q^{1/2}, 0, +1/q^{1/2}\}$ с соответствующими вероятностями $\{q/2, 1-q, q/2\}$ и др. [11, 16–18]. Размерность d в леммах (D)JL зависит от c. Отметим, что для выполнения леммы DJL следует умножать $\mathbf{Rz}$ с такими $\mathbf{R}$ на $1/\sqrt{d}$ либо получать $\mathbf{R}$ таким же масштабированием (субгауссовых) с.в.

Рассмотрим связь JLT с матрицами, которые имеют свойство ограниченной изометрии (Restricted Isometry Property, RIP) [18–20]: для любого k-разреженного вектора (т.е. вектора с не более чем k ненулевыми компонентами) умножение на RIP-матрицу сохраняет квадрат евклидовой нормы с искажением $1 \pm \varepsilon$. Такие RIP-матрицы используют в задачах «сжатого измерения» (compressed sensing) (см. ссылки в [18–20]). Для JLT-матриц с большой вероятностью выполняется RIP (с другими константами и для векторов с разреженностью k до некоторой

оптимальной). Например, гауссовы и рademахеровы случайные матрицы $\mathbf{R}$ являются RIP при $d = O(\varepsilon^{-2} k \log(D/k))$ [20], и наоборот, матрица $\mathbf{RD}_R$ (где $\mathbf{R}$ — это RIP$(k, \varepsilon/4)$-матрица, а $\mathbf{D}_R$ — диагональная рademахерова матрица) с большой вероятностью JLT (с субоптимальным d, $d = O(\varepsilon^{-2} k \log D)$ при $N \le 2^k$) [21, 22]. Недавно полученные результаты по RIP приведены в [20–23].

Помимо RIP, аналоги леммы JL существуют и для других бесконечных (непрерывных) множеств с некоторыми ограничениями: многообразий, линейных подпространств, объединений подпространств и др. (см. [18, 24] и ссылки к ним). Так, пусть непрерывное множество S имеет гауссову (среднюю) ширину (Gaussian width) ω, которая определяется как $\omega(S) = \mathrm{E}\{\sup_{x \in S} \langle \mathbf{r}, \mathbf{x} \rangle\}$, где $\mathbf{r} \sim \mathrm{Norm}(\mathbf{0}, \mathbf{I}_D)$, E — математическое ожидание (м.о). Для гауссовой i.i.d. $\mathbf{R}$ при $d = O(\omega^2(S)/\varepsilon^2)$ для S выполняется аналог леммы JL с аддитивным искажением $\pm\varepsilon$ (3) [24–27]. Для матриц с i.i.d. субгауссовыми элементами подобный результат получен в [28], см. также [18]. Отметим, что $\omega(S) \le (2 \log |S|)^{1/2}$ [27].

Для матриц с RIP для различных уровней искажения ε и разреженности k при их умножении на $\mathbf{D}_R$ также с большой вероятностью выполняется аддитивный аналог леммы JL для непрерывных ограниченных множеств S при d, зависящим от $\omega^2(S)$ [27]. Существование RIP-матриц с быстрым умножением (разд. 3) позволяет ускорить такое случайное проецирование.

2.2. Дерандомизация случайного проецирования. Условия леммы JL предполагают необходимость рандомизированных вложений для снижения размерности, так как для детерминированной матрицы при $d < D$ существует бесконечно много $\mathbf{x} : \mathbf{Rx} = 0$ (векторы из нуль-пространства $\mathbf{R}$). Поэтому дерандомизация JLT состоит в поиске таких классов JLT-матриц, которые можно сгенерировать (либо выбрать из всех «готовых» матриц некоторого распределения) по минимальному числу случайных битов (см. [13, 29, 30] и ссылки к ним). Это важно для приложений, где память ограничена. Так, генерация матрицы в [30] требует всего $d = O(\log(1/\delta) \log D)$ случайных битов (подразд. 3.3). Однако существуют процедуры конструирования явных JLT-матриц для заданного множества N векторов [31, 32].

2.3. Дисперсии оценок. Аналоги леммы JL для скалярного произведения, угла и вложений в L_1. Лемма JL дает вероятностные гарантии худшего случая на $1 \pm \varepsilon$-искажения евклидова расстояния. Дисперсия V оценки меры расстояния/сходства является мерой неточности вложения в среднем (и в ряде случаев позволяет получить оценки худшего случая при известном распределении ошибки). Источник случайности оценок при случайном проецировании — различные реализации $\mathbf{R}$. Дисперсия V оценки $||\mathbf{x} - \mathbf{y}||_2^2$ по d-мерным скетчам [33, 34]:

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

где ρ — с.в. (элемент $\mathbf{R}$). Для тернарных матриц с i.i.d. элементами из $\{-1/q^{1/2}, 0, +1/q^{1/2}\}$ с вероятностями $\{q/2, 1-q, q/2\}$, $\mathrm{E}\{\rho^4\}/\mathrm{E}^2\{\rho^2\} = 1/q$ [33], а для бинарных из $\{0,1\}$: $1/(q - q^2) - 3$ [34]. Для гауссовых i.i.d. матриц V дает (4) с $\mathrm{E}\{\rho^4\}/\mathrm{E}^2\{\rho^2\} = 3$ [35, 33].

Для оценки скалярного произведения $\langle \mathbf{x}, \mathbf{y} \rangle$ [33, 34]

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

Учет значений норм $||\mathbf{x}||_2, ||\mathbf{y}||_2$ позволяет повысить точность оценок [33].

Из $||\mathbf{x}-\mathbf{y}||_2^2 = ||\mathbf{x}||_2^2 + ||\mathbf{y}||_2^2 - 2\langle \mathbf{x}, \mathbf{y}\rangle$ для JLT-матриц следует существование аналогов лемм (D)JL для $\langle \mathbf{x}, \mathbf{y}\rangle$ и косинуса угла $\cos(\mathbf{x}, \mathbf{y})$ между единичными $\mathbf{x}, \mathbf{y}$. Однако здесь искажение $\pm\varepsilon_a$ аддитивно (3), причем $\varepsilon_a = \varepsilon ||\mathbf{x}||_2 ||\mathbf{y}||_2$ [36, 29, 37]. Лемма с оптимальными вероятностями δ (как в лемме JL) приведена в [37], где также показано, что для $1\pm\varepsilon$-искажения δ зависит от $\cos^2(\mathbf{x}, \mathbf{y})$. Быстрая оценка $\langle \mathbf{x}, \mathbf{y}\rangle$ может быть полезна для оценки ядерных сходств (разд. 7).

Для преобразований, сохраняющих евклидовы расстояния с $1\pm\varepsilon$-искажением, существует аналог лемм JL для оценки угла между единичными векторами с $\pm\varepsilon_a$-искажением [29, 38]. Сохранение угла с искажением $1\pm\varepsilon$ требует увеличения числа сохраняемых расстояний, т.е. размерности d (и/или ε) [39].

Для вложения N векторов из L_2 в L_1 условия (аналогов) лемм (D)JL с искажением $1\pm\varepsilon$ выполняются с незначительными изменениями констант (для гауссовой случайной i.i.d. матрицы [26, 40]). Использование разреженных матриц (с малой долей ненулевых элементов) требует «гладкости» $\mathbf{x}$ [41, 11] (для разреженных гауссовых матриц [41, 11], для тернарных матриц [11], для вложений в L_2 см. подразд. 3.1).

Вложение из L_2 в L_1 непрерывных ограниченных множеств с $\omega(S)$ с помощью гауссовой матрицы возможно с искажением $\pm\varepsilon_a$ при $d = O(\omega^2(S)/\varepsilon_a^2)$ [26, 42], а вложение всего D-мерного пространства L_2^D в L_1^d с искажением $1\pm\varepsilon$ — только для $d = O(D)$ [3, 4]. В разд. 4 приведены результаты, свидетельствующие о невозможности снижения размерности для L_s, $s \neq 2$. Преимущества и недостатки вложений векторов евклидовых пространств рассмотрены в подразд. 9.1.

3. УСКОРЕНИЕ СЛУЧАЙНОГО ПРОЕЦИРОВАНИЯ

К недостаткам случайного проецирования JLT относится большая вычислительная сложность $O(Dd)$ умножения вектора на матрицу при прямой реализации (но см. [43]). Время уменьшается до $O(\mathrm{nnz}(\mathbf{x})d)$ для разреженного $\mathbf{x}$ с числом ненулевых компонентов $\mathrm{nnz}(\mathbf{x})$. Для произвольных $\mathbf{x}$ ускорения JLT можно достичь применением специальных матриц: разреженных либо неразреженных, но позволяющих быстрое умножение.

3.1. Случайные i.i.d. разреженные матрицы. Время умножения плотного $\mathbf{x}$ проходом по ненулевым элементам матрицы (с долей или вероятностью q ненулевых элементов) составляет $O(Dqd)$, а k-разреженного $\mathbf{x}$ — $O(kqd)$ (например, [44]). Особенно эффективно оперирование матрицами с элементами из $\{-1, 0, +1\}$ [33, 45–47] и $\{0, +1\}$ [34, 48]. Дисперсии оценки евклидова расстояния и скалярного произведения из [34] даны в (4), (5). В [33] приведена скорость их сходимости к дисперсии гауссовых проекций, а в [33, 48] — скорость сходимости распределения элементов выходного вектора к гауссову.

Для разреженных матриц и разреженных векторов $\mathbf{x}$ число ненулевых произведений $x_i r_i$ (при вычислении $\langle \mathbf{x}, \mathbf{r}\rangle$ в $\mathbf{Rx}$) может быть недостаточным для необходимой в лемме DJL (см. подразд. 2.1) концентрации $||\mathbf{Rx}||_2$ вокруг $||\mathbf{x}||_2$. Например, вектор с $||\mathbf{x}||_2 = 1$ может содержать всего один единичный компонент. Поэтому лемма DJL для случайных i.i.d. разреженных матриц [41, 11] требует ограничения разреженности векторов (что неявно задается как $||\mathbf{x}||_\infty / ||\mathbf{x}||_2 \le \alpha$, $\alpha \in [1/D^{1/2}, 1]$, и близко к $1/D^{1/2}$) и матрицы ($q = C_0\alpha^2 \log(D/\varepsilon\delta)$). При этом $d = C\varepsilon^{-2}\log(4/\delta)$ и м.о. числа c ненулевых компонентов в столбце $\mathrm{E}\{c\} = \widetilde{\Omega}(\alpha^2/\varepsilon^2)$, где $\widetilde{\Omega}(f)$ обозначает функцию вида $f\log^{\Omega(1)}(f)$.

3.2. Матричные конвейеры для быстрого преобразования JL. Для ускорения JLT (с возможностью применения для разреженных векторов) используют различные матричные «конвейеры» (pipeline) — последовательность матриц, умноже-

ние которых на вектор быстро вычислимо, а для результирующих векторов обеспечивается выполнение леммы JL с близкими к оптимальным параметрами.

Матричный конвейер из [41] называется Fast Johnson–Lindenstrauss Transform (FJLT). Для достижения нужного α вектора $\mathbf{x}$ осуществляется его предобработка (precondition). В [41] это обеспечивается случайным вращением $\mathbf{x}$ посредством $\mathbf{HD}_R\mathbf{x}$, где $\mathbf{H}$ — (ортогональная) матрица Адамара:

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

Умножение вектора на $\mathbf{H}$ выполняется за время $O(D \log D)$. При этом с большой вероятностью достигается $\alpha = O(\sqrt{d/D}) \sim O(\sqrt{\log N / D})$, что позволяет использовать разреженную i.i.d. матрицу $\mathbf{G}$ (в [41] гауссову с оптимальным d и с $q \sim d^2/D$ и $\mathrm{nnz}(\mathbf{G}) = O(d^3)$). В результате получают FJLT-преобразование $\mathbf{GHD}_R\mathbf{x}$. Время выполнения $O(D \log D + d^3)$. Дальнейшее развитие [49–51] и применение RIP-матриц (см. подразд. 2.1) позволило уменьшить и затем убрать зависимость времени от d за счет увеличения d по сравнению с оптимальным в лемме JL. Так, в [20] достигнуто время $O(D \log D)$ для $\mathbf{x}$ с $\mathrm{nnz}(\mathbf{x}) \le D/\mathrm{poly}\log D$ при $d = O(\varepsilon^{-2} \log N \log^3 D)$.

Очень простыми матрицами, обеспечивающими время умножения $O(D \log D)$, являются теплицевы и циркулянтные матрицы. Теплицевы матрицы имеют одинаковые элементы на диагоналях (задаются $D + d - 1$ числами). В циркулянтных матрицах (обозначим их $\mathbf{C}$) строки получают циклическим сдвигом первой (т.е. требуется D чисел). Для случайного проецирования элементы строки обычно i.i.d. гауссовы или рademахеровы с.в. ($\mathrm{vec}(\mathbf{D}_G)$ или $\mathrm{vec}(\mathbf{D}_R)$). Такие матрицы при надлежащем выборе параметров являются RIP с большой вероятностью [52], т.е. $\mathbf{CD}_R$ есть JLT (см. подразд. 2.1 и [21]). Совершенствование анализа JLT-конвейера $\mathbf{CD}_R$ позволило улучшить требуемое d с $O(\varepsilon^{-2} \log^3 N)$ [53] до $O(\varepsilon^{-2} \log^2 N)$ [54] и даже до $O(\varepsilon^{-2} \log^{(1+\eta)} N)$ (см. [55, 56] и ссылки к ним).

Подобные быстрые конвейеры используются и для быстрой реализации RIP-преобразований (см. подразд. 2.1), при получении бинарных скетчей [1], аппроксимации ядер (разд. 7) и линейной части слоев нейронных сетей [57–60]. Отметим, что элементы матрицы произведения матриц конвейера не являются (гауссовыми) i.i.d., что затрудняет анализ таких «структурированных» случайных матриц.

3.3. Разреженное преобразование JL. Во всех вариантах FJLT не используется возможная разреженность $\mathbf{x}$ (наоборот, зачастую осуществляется «уплотнение»). Это приводит к тому, что для вектора с одним ненулевым компонентом (например, в режиме потоковой обработки (см. подразд. 1.3)) время $O(D \log D)$ модификации скетча намного больше «наивного» $O(d)$. Кроме того, во многих приложениях применяют разреженные векторы (представления текстов словами, рекомендации или покупки пользователей и т.д.).

Чтобы ускорить умножение разреженных векторов на разреженные матрицы и преодолеть ограничение на c (см. подразд. 3.1), в качестве элементов матрицы используют не i.i.d. с.в. Так, в [61] предложен так называемый hashing-trick: несмещенную оценку скалярного произведения находят по скетчам, полученным с использованием хэш-функций $h:[D] \to [d]$ и $g:[D] \to \{-1, +1\}$, где $[n]$ обозначает $\{1, 2, \ldots, n\}$. Компонент скетча формируют сложением отображенных в него компонентов исходного вектора, умноженных на соответствующие им значения из $\{-1, +1\}$. Это идентично умножению на матрицу с ровно одной $+1$ или -1 в столбце (случайно расположенной). При этом дисперсия оценки такая же, как у матриц с i.i.d. с.в. из $\{-1, +1\}$ [61, 62]. Для достижения нужного α используется

простое детерминированное «уплотнение» **x** с помощью c-кратного «размножения» его компонентов и их деления на $c^{1/2}$, что обеспечивает сохранение $||x||_2$ и уменьшение $||x||_\infty$ в $c^{1/2}$ раз. При этом хэширование модифицируется как $h:[cD]\to[d]$ и $g:[cD]\to\{-1,+1\}$. Результирующее преобразование **x** можно реализовать умножением на псевдослучайную матрицу $d\times D$, в столбцах которой от одного до c ненулевых элементов (вследствие возможных коллизий c хэшей одного и того же компонента **x**).

В результате анализа рассмотренной схемы как JLT [63] получено $c=O(\varepsilon^{-1}\log(1/\delta)\log^2(d/\delta))$ для $d=O(\varepsilon^{-2}\log(1/\delta))$. В [64] показано, что достаточно, чтобы $c=O(\varepsilon^{-1}\log(1/\delta)\log(d/\delta))$; некоторое улучшение этого папаметра достигнуто в [65]. Для дальнейшего уменьшения c в [30] предложено использовать в столбце матрицы **R** ровно c ненулевых элементов из $\{-1,+1\}$. В одном из вариантов столбец матрицы **R** разбивают на c непрерывных блоков размерности d/c и в каждом случайно размещают один элемент: -1 или $+1$. Это позволило улучшить разреженность до $c=\Theta(\varepsilon^{-1}\log(1/\delta))\sim\varepsilon^{-1}\log|S|$ при оптимальном $d=O(\varepsilon^{-2}\log(1/\delta))$. Таким образом, доля ненулевых элементов матрицы и ускорение вычисления этого разреженного преобразования JL (SJLT) равны ε. Такое c близко к оптимальному $c=\Omega(\varepsilon^{-1}\log(1/\delta)/\log(1/\varepsilon))$ [66].

В [24] исследуется выполнение аналогов леммы JL с искажением $1\pm\varepsilon$ евклидова расстояния для SJLT и различных множеств единичных векторов с некоторыми ограничениями на геометрию посредством «параметра сложности».

Таким образом, структурированные (т.е. не гауссовы i.i.d.) матрицы позволяют ускорить умножение, уменьшить затраты памяти на хранение матрицы и количество требуемых случайных чисел, упростить алгоритмическую реализацию. Сравнительное экспериментальное исследование алгоритмов снижения размерности на основе леммы JL для различных матричных конвейеров приведено в [67].

4. ВЛОЖЕНИЯ И СКЕТЧИ ДЛЯ ОЦЕНКИ НЕЕВКЛИДОВЫХ РАССТОЯНИЙ МИНКОВСКОГО

Для других расстояний L_s, $s\neq 2$, снижение размерности $L_s^D\to L_s^d$ линейным преобразованием с искажением (постоянным для векторов размерности d и не зависящем от D) в общем (худшем) случае невозможно (для L_1 см. [68, 69]). Доказательства основаны на демонстрации наборов N векторов в пространстве размерности $D=N$, вложение которых с заданным искажением требует большой размерности.

Так, для линейного вложения N векторов из L_s в L_s^d при $1\le s\le\infty$ мультипликативное искажение A (2) составляет не менее $A=\Omega((N/d)^{|1/s-1/2|})$ [70]. Для L_1 это означает, что $d\ge CN/A^2$. Это справедливо и для вложения L_1 в любое L_s [70]. Более сильный результат для L_1 утверждает, что $d\ge N^{\Omega(1/A^2)}$ [69, 68]. Для малых искажений $1+\varepsilon$ нижняя граница $d\ge N^{1-O(1/\log(1/\varepsilon))}$ [71], а верхняя граница $d\le O(N/\varepsilon^2)$ [72].

Однако и преобразование метрики без снижения размерности может быть полезным (см. подразд. 1.3). Отметим, что для $1\le t\le s\le 2$ все пространство L_s^D можно вложить в L_t^{CD} с искажением $1+\varepsilon$, причем $C=C(s,t,\varepsilon)=O(\varepsilon^{-2}\log(1/\varepsilon))$ не слишком велико [73, 74, 3, 4] (а также см. подразд. 3.1 и [4] для $L_2\to L_1$). При $C>1$ возможно конструирование явного вложения. Однако характеристики таких вложений намного хуже, чем у рандомизированных [4].

Таким образом, для L_s, $s\neq 2$, невозможно линейное снижение размерности произвольного множества N векторов с постоянным искажением в худшем слу-

чае. Для преодоления этого ограничения используют: ограничения на множества векторов (подразд. 4.1); скетчи с оценкой L_s ($0< s \le 2$) не по расстоянию в целевом пространстве (подразд. 4.2). Отметим, что для работы с расстояниями L_s при $s>2$ эффективные скетчи [75] (постоянной размерности и с постоянным искажением, не зависящими от D) в принципе невозможны, требуемая размерность $d=\Omega(D^{1-2/s})$ [76].

4.1. Вложения подмножеств L_1. Для k-разреженных векторов из L_1 возможно $1\pm\varepsilon$-вложение в $d \ge Ck\log(D/k)/\varepsilon^2$ (с большой вероятностью) с помощью L_1-RIP-матриц (в частности, промасштабированных бинарных, содержащих $C\varepsilon^{-1}\log(D/k)$ единиц в столбце) (см. [77] и Prop. 1 в [78]). Подобные результаты для $1 \le s \le \infty$ получены в [79].

Подход к вложению в L_1 s-блоковых норм [78] использует поэлементное произведение матриц, одна из которых бинарная с фиксированным числом случайно расположенных единиц в каждом столбце, другая — гауссова. При изменении s удается воспроизвести известные результаты для вложений $L_2 \to L_1$ и $L_1 \to L_1$ для подмножеств векторов с определенными свойствами.

Нелинейные вложения с искажением $1+\varepsilon$ в ограниченном диапазоне расстояний из L_s в L_t, $1 \le t \le s \le 2$, при целевой размерности $d=O(\log N)$, которая также зависит от значений диапазона и ε, t, s, предложены в [80]. Одномерное вложение $\mathbf{x}$ осуществляется как $\sin(2\langle \mathbf{x}, \mathbf{r}\rangle / a + \xi)$ с масштабированием, где $\mathbf{r}$ — случайный вектор из s-устойчивого распределения (подразд. 4.2), a — значение, связанное с величиной диапазона, $\xi \sim \mathrm{Unif}[0, 2\pi]$. Такие вложения (с ограниченным диапазоном) полезны для поиска по сходству, кластеризации и др.

4.2. Скетчи для расстояний L_s ($0< s \le 2$) на основе устойчивых случайных проекций. Еще одним подходом к быстрой оценке расстояний Минковского L_s ($0< s \le 2$) с малым искажением является создание скетчей малой размерности $d=O(\log N)$, с которыми в отличие от вложений вместо L_s используются другие оценки исходных расстояний [81]. Для них существуют аналоги леммы JL с искажением $1\pm\varepsilon$ для вычисленных по скетчам оценок расстояний.

Скетчи для L_s ($0< s \le 2$) на основе устойчивых случайных проекций формируют как $\mathbf{Rx}$ ($\mathbf{R}$ — случайная матрица с i.i.d. элементами из s-устойчивого распределения [81]). Например, для L_1 применяют 1-устойчивое распределение Коши. Отметим, что 2-устойчивым является распределение Гаусса.

Для оценки L_s по скетчам используют варианты медианных оценок абсолютных значений разности компонентов скетчей [81]. Для повышения точности применяют следующие оценки: по медиане, геометрическому среднему, гармоническому среднему, дробных степеней, оптимальные квантильные, максимального правдоподобия с коррекцией смещения (bias-corrected) (см. [82, 83] и ссылки к ним). Преимущества и недостатки скетчей устойчивым случайным проецированием описаны в подразд. 9.2.

5. ОЦЕНКИ ЛИНЕЙНЫХ СУММИРУЮЩИХ СТАТИСТИК ПО СКЕТЧАМ, ПОЛУЧЕННЫМ СЭМПЛИРОВАНИЕМ

Отбор подмножества элементов исходного представления объекта называют сэмплированием. Целью сэмплирования обычно является получение представления исходного объекта, по которому можно оценить некоторые его характеристики. Для быстрой оценки мер расстояний/сходств с использованием сэмплирования применяют, главным образом, представления исходных объектов, которые можно рассматривать как векторы. Представимы векторами и полученные сэмплированием скетчи. (Часто используют представление в виде пар (ID, value) с value $\ne 0$, где компоненты с одинаковым ID соответствуют один другому. Такие представления легко развернуть в обычные векторы.)

Рассмотрим методы случайного сэмплирования (см. [84, 85] и ссылки к ним). В простом случайном сэмплировании с замещением (simple random sampling with replacement) каждый компонент отбирается в скетч с одинаковой вероятностью (и может быть отобран несколько раз). Число отобранных в скетч компонентов выборки фиксировано. При сэмплировании с замещением PPS (probability proportional to size) вероятность отбора пропорциональна весу компонента (например, его величине). Для данных с «тяжелыми хвостами» (где бóльшая часть веса сконцентрирована в малом числе компонентов с большими значениями) это может привести к тому, что в скетче окажутся одни и те же тяжелые компоненты, а другие будут плохо представлены.

В сэмплировании без замещения компонент выбирается не более одного раза, что позволяет получать более точные оценки. Различают сэмплирование с одинаковой вероятностью (бернуллиево) и с разной вероятностью, например, пропорциональной величине компонентов (пуассоново). Оба типа сэмплирования отбирают нефиксированное число компонентов. Если его фиксировать, то бернуллиево сэмплирование становится простым случайным без замещения, а пуассоново — условным.

Бернуллиево сэмплирование легче анализировать, но оценки имеют меньшую точность для данных с тяжелыми хвостами. Пуассоново сэмплирование позволяет находить более точные оценки, но их получение по скетчам нетривиально; также используются ограничения на данные, например, неотрицательность компонентов векторов, что соответствует взвешенным множествам.

Случайное сэмплирование представления объекта не учитывает информации о других объектах (т.е. забывчивое). Однако при сэмплировании различных объектов можно использовать как разные наборы случайных чисел (независимое сэмплирование), так и одинаковые (скоординированное сэмплирование, coordinated). Для оценки сходства в основном применяется скоординированное сэмплирование. Дисперсии оценок той или иной величины по скетчам, полученным сэмплированием, уменьшаются с ростом размерности скетча d.

5.1. Скетчи, получаемые простым случайным сэмплированием без замещения. При простом случайном сэмплировании без замещения скетчи размерности d получают случайной перестановкой исходных векторов (для устранения потенциально имеющейся в них структуры) и отбором их первых d компонентов. Отметим, что такой же скетч даст умножение входного вектора на соответствующую бинарную матрицу с одной единицей в каждой строке.

Пусть мера расстояния/сходства sim исходных векторов определяется как $\mathrm{sim}\,(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{D} \mathrm{sim}_i\,(x_i, y_i)$, т.е. является линейной суммирующей статистикой (linear summary statistics), например, скалярное произведение, квадрат евклидова расстояния, расстояние χ^2 (подразд. 8.1) и др. Тогда несмещенную оценку $\mathrm{sim}^*\,(\mathbf{x}, \mathbf{y})$ по значению $\mathrm{sim}\,(\hat{\mathbf{x}}, \hat{\mathbf{y}})$, вычисленному по скетчам $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ размерности d, и ее дисперсию V получают [86] как

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

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

Для $D >> d$, а также для $\sum_{i=1}^{D} \mathrm{sim}_i^2\,(x_i, y_i) >> \mathrm{sim}^2\,(\mathbf{x}, \mathbf{y}) / D$ (т.е для векторов с тяжелыми хвостами) значение V велико.

5.2. Скетчи для разреженных векторов, получаемые условным случайным сэмплированием. Для сильно разреженных векторов специализированные методы сэмплирования дают более точные оценки исходных мер расстояния/сход-

ства при тех же затратах памяти на скетч. Условное случайное сэмплирование (Conditional Random Sampling, CRS) [86] формирует скетч отбором заданного числа первых ненулевых компонентов вектора после случайной перестановки (нулевые компоненты не используются в скетче). При вычислении (6) применяется [86] $d = \min\{\max \mathrm{ID}(\hat{\mathbf{x}}) - 1, \max \mathrm{ID}(\hat{\mathbf{y}}) - 1\}$, где $\max \mathrm{ID}(\hat{\mathbf{x}})$ — максимальный ID компонент в скетче $\hat{\mathbf{x}}$. Пусть $d_{\hat{\mathbf{x}}}$ — число компонентов в скетче $\hat{\mathbf{x}}$, тогда дисперсия

$$\mathrm{V}\{\mathrm{sim}^*(\mathbf{x}, \mathbf{y})\} \approx D/(D-1)(\max\{\mathrm{nnz}(\mathbf{x})/(d_{\hat{\mathbf{x}}} - 1), \mathrm{nnz}(\mathbf{y})/(d_{\hat{\mathbf{y}}} - 1)\} - 1) \times$$

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

Сравнивая эту дисперсию с (7), видим, что в отличие от применения обычного сэмплирования она меньше приблизительно в $D/\mathrm{nnz}(\mathbf{x})$ раз. Для данных с тяжелыми хвостами дисперсия остается большой вследствие возможности пропуска «тяжелых» компонентов.

5.3. Скетчи, получаемые взвешенным сэмплированием. Для данных с тяжелыми хвостами сэмплирование (без замещения) должно давать приоритет компонентам векторов с большими значениями. В приоритетном (priority) сэмплировании [84] (последовательном пуассоновом) скетч фиксированной размерности d для векторов с положительными компонентами (обозначим их $\mathbf{x} > 0$) формируют следующим образом. Каждому компоненту назначают приоритет $\beta_i = x_i / r_i$, $i \in [D]$, где $r_i \sim \mathrm{Unif}(0, 1]$ (в алгоритме получают хэшированием номера компонента [87]), и β_i упорядочивают по величине. Определяют порог $\beta = \beta_{d+1}$. Скетч формируют как $\hat{x}_i = \max\{x_i, \beta\}$, если $\beta_i > \beta$, и 0 в протвном случае. Отметим, что приоритетное сэмплирование пригодно для потоковой обработки и первоначально применялось для оценки суммы компонентов вектора. В [87] с помощью специального варианта таких скоординированных скетчей рассмотрена оценка $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y})$ (обобщенный коэффициент Жаккара для $\mathbf{x}, \mathbf{y} > 0$) [88, 89]

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

Показано, что при больших d использование всего лишь 2-независимых хэш-функций позволяет достичь малого смещения (и дисперсии) оценки $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y})$. Отметим связь $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y})$ с $||\mathbf{x} - \mathbf{y}||_1$ [89] через

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

и обобщение $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y})$ на вещественные векторы [90] заменой $x < 0$ парой компонентов $[0 \; -x]$, а $x > 0$ — парой $[x \; 0]$.

Скетчи, получаемые сэмплированием, для оценки расстояний L_s (в частности, L_1 и L_2) между неотрицательными векторами рассмотрены в [85]. Кроме приоритетного сэмплирования, для формирования скетчей используют пуассоново сэмплирование PPS, где компонент включают в скетч, если $x_i > r_i \beta$. Значение β задается или выбирается с использованием $\mathrm{E}\{d\} = \sum_{i=1}^{D} \min(1, r_i / \beta)$. Рассмотрены два типа сэмплирования: независимое (r_i различные для разных скетчей) и скоординированное (r_i одинаковые для разных скетчей). Для оценки $(L_s)^s$, помимо отобранных компонентов $\{i, x_i\}$ с $\beta_i > \beta$, $i \in [D]$, используют их r_i и β_{d+1}, β_d. Оценка проводится по соответствующим компонентам скетчей, зависит от s в $(L_s)^s$, типа сэмплирования и является нетривиальной [85]. Общий подход к оценке других мер расстояний/сходств по скетчам, полученным по результатам взвешенного сэмплирования, представлен в [91].

Преимущества и недостатки скетчей, получаемых сэмплированием, описаны в подразд. 9.3.

6. ОЦЕНКА РАССТОЯНИЙ МЕЖДУ НЕВЕКТОРНЫМИ ДАННЫМИ

В подразд. 6.1 рассмотрены универсальные методы вложений для оценки любых исходных расстояний, а в подразд. 6.2 — специализированные методы вложений для некоторых невекторных расстояний.

6.1. Формирование векторных представлений на основе расстояний. Методы формирования векторных представлений объектов на основе их расстояний до некоторых выделенных («опорных») объектов (ОО) универсальны, так как они не требуют доступа к исходным представлениям объектов и применимы для различных исходных расстояний. Поэтому пространства исходных представлений объектов могут быть векторными, метрическими, неметрическими.

В методе классического многомерного шкалирования (MDS) [3] исходную матрицу $N \times N$ расстояний dist между (под)множеством ОО базы подвергают «двойному центрированию», превращая в матрицу сходств $\mathbf{K}$, $K_{ij} = \text{dist}^2(x_0, x_i) + + \text{dist}^2(x_0, x_j) - \text{dist}^2(x_i, x_j)$, $i, j \in [N]$. Если в исходной матрице евклидовы расстояния, то $\mathbf{K}$ есть PSD и ее можно рассматривать как матрицу ядра (разд. 7). Тогда с помощью PCA из $\mathbf{K}$ формируют векторные представления объектов. Это вложение является изометрией для N исходных объектов. Такое MDS является (слабо)забывчивым, так как возможно и приближенное вложение нового объекта x методом Нистрема (Nystrom) [92] как $\boldsymbol{\psi}^*(x) = \boldsymbol{\Lambda}^{-1/2}\mathbf{U}^{\mathrm{T}}\boldsymbol{\kappa}_x$, где $\mathbf{U}$ — матрица собственных векторов (в столбцах) в результате PCA (собственного разложения матрицы ядра $\mathbf{K}$ для N объектов), $\boldsymbol{\Lambda}^{-1/2} = \text{diag}\,(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots)$, $\boldsymbol{\kappa}_x = (\kappa(x, x_1), \ldots \ldots, \kappa(x, x_N))^{\mathrm{T}}$ — значения ядерных сходств нового объекта с ОО, которые можно получить из вектора соответствующих расстояний.

Примерами приближенных забывчивых вложений на основе расстояний являются FastMap, MetricMap, SparseMap (см. [93] и ссылки к ней). Вложения FastMap и MetricMap в [94] рассматриваются как относящиеся к классу MDS, использующие разновидности или обобщения метода Нистрема, причем с потерей точности.

В изометрическом вложении Фреше [3] из конечного метрического пространства с N объектами и dist в L_∞ координата i целевого векторного N-мерного пространства определяется как расстояние объекта y (одного из ОО) до i-го ОО: $\mathrm{f}_i(y) = \text{dist}(y, x_i)$, $i \in [N]$. Изометрическое вложение Фреше не является забывчивым.

Сжимающие (contraction) вложения, т.е. не увеличивающие исходных расстояний, важны для поиска по сходству, так как позволяют получить точные результаты поиска [93]. Например, сжимающим является расстояние L_s для L_t при $s > t > 0$ (без изменения векторных представлений), а также вложение Фреше для новых объектов и L_∞ (либо с другим расстоянием L_s при надлежащем нормировании [93]).

Применение векторных представлений на основе расстояний для классификации и других задач распознавания образов приведено в [95]. Для поиска по сходству в [96] применяют векторное представление объекта, компоненты которого — номера ОО, упорядоченные по величине сходств/расстояний до объекта.

Недостатками этих методов являются зачастую эвристический характер выбора ОО, трудоемкость вычисления сложных исходных расстояний (например, расстояния редактирования для графов [97]), отсутствие аналитических оценок искажений.

6.2. Вложения объектов со специальными метриками. Конструирование быстрых и забывчивых алгоритмов формирования векторов для оценки расстояний невекторных исходных объектов со специфицированным и минимальным искажением является сложной задачей. Поэтому такие алгоритмы обычно специализированы для конкретных исходных представлений и типов расстояний.

Несмотря на искажение, размерность и время получения, увеличивающиеся с размерностью исходных представлений, востребованы вложения специализи-

рованных метрик (расстояний между невекторными исходными объектами) в L_1, а также в $(L_2)^2$, L_∞ и др. [75, 98]. Это обусловлено существованием для L_s эффективных скетчей (см. разд. 4, для L_1 см. подразд. 4.2). Таким образом, специализированные расстояния между невекторными исходными объектами вкладывают, например, в L_1, которое можно оценить по скетчу малой размерности с небольшим дополнительным искажением (см. подразд. 4.2). Кроме того, для векторов разработаны алгоритмы быстрого поиска по сходству (подразд. 8.3).

Для символьных последовательностей (строк) часто используют расстояние редактирования Левенштейна ($\mathrm{dist}_{\mathrm{edit}}$) [99, 100], равное минимальному числу элементарных операций редактирования символов строки, необходимых для преобразования одной строки в другую. Элементарными операциями являются вставка, удаление, замена символа в определенной позиции. Сложность вычисления с использованием динамического программирования квадратичная от длины строк n.

Вложения $\mathrm{dist}_{\mathrm{edit}}$ в L_1 в основном рассматривают строки $\{0,1\}^n$ и используют в качестве компонентов скетчей некоторые (непрерывные) подстроки исходных строк, т.е. являются нелинейными. Отметим, что для размера алфавита 4 [101] и даже 2 [102] невозможно точно вычислить $\mathrm{dist}_{\mathrm{edit}}$ за время $O(n^{2-\varepsilon})$, если выполняется strong exponential time hypothesis [101].

Анализируют мультипликативное искажение A (2). Для варианта $\mathrm{dist}_{\mathrm{edit}}$ с дополнительной возможностью перемещения блоков получены [103] вложения за почти линейное время с искажением $\widetilde{O}(\log n)$. Однако для классического $\mathrm{dist}_{\mathrm{edit}}$ подобных результатов долгое время достичь не удавалось.

В [104] получено вложение классического $\mathrm{dist}_{\mathrm{edit}}$ с искажением $\Omega(n^{1/2})$ и временем вычисления $O(n^{3/2})$, а в [6] — с искажением $n^{1/3+o(1)}$ за линейное время или $n^{\varepsilon/3+o(1)}$ за время $O(n^{2-\varepsilon})$ (но не в L_1, а в пространство строк уменьшенной длины).

Искажение $2^{\widetilde{O}(\sqrt{\log n})}$ вложения в L_1 (что меньше n^ε для любого $\varepsilon > 0$) показано в [105], однако неизвестно, возможно ли вычислить это вложение за субквадратичное время. В [106] получена такая же аппроксимация, но за время $n^{1+o(1)}$, причем используется не только вложение в L_1, но и другие, незабывчивые вложения. В [107] с использованием сэмплирования одной из строк достигнуто искажение $(\log n)^{O(1/\varepsilon)}$, но за время $n^{1+\varepsilon}$, что хуже, чем в [106].

Нижняя граница $\Omega(\log n)$ на мультипликативное искажение расстояния редактирования строк $\{0,1\}^n$ при вложении в L_1 показана в [108], а нижняя граница $\Omega(n)$ для скетчей, полученных случайным линейным проецированием, — в [109].

Вложения других расстояний описаны в [3, 98] (см. также ссылки к ним и на них), однако не все они являются забывчивыми, даже по ослабленному определению (см. подразд. 1.2).

7. ЯДЕРНЫЕ СХОДСТВА И ИХ АППРОКСИМАЦИЯ

Специальным видом функции сходства является ядерная функция (ядро) $\kappa(x, y)$ [110]. Это непрерывная, вещественная, симметричная, положительно полуопределенная (PSD) функция. Одно из определений PSD $\kappa(x, y)$ — существование (возможно, неявного) преобразования $\varphi : X \to H$ исходных объектов x, y в векторы $\varphi(x)$, $\varphi(y)$ во «вторичном» или «признаковом» гильбертовом пространстве H (возможно, бесконечной размерности) такого, что $\kappa(x, y) = \langle \varphi(x), \varphi(y) \rangle$.

Ядерное сходство вычисляется по исходным представлениям объектов некоторого типа (векторы, последовательности, графы и др.) с помощью ядерной

функции. Сложность вычисления ядер зависит от конкретного типа ядра и обычно полиномиальная. Примерами ядер для векторов $\mathbf{x}, \mathbf{y}$ являются $\kappa(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$ — линейное ядро, полиномиальное

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

и гауссово RBF

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

Другим примером являются ядра для структурированных объектов (графов и др.), основанные на их разбиении на подструктуры со своими локальными ядерными сходствами [111–114].

Ядерные алгоритмы зависят только от $\kappa(x, y)$, однако обычно требуют вычисления (и использования) N^2 ядерных сходств между N объектами (т.е. матрицы ядра $\mathbf{K}$). Для больших N это зачастую невозможно.

Для быстрой оценки элементов $\mathbf{K}$ используют низкоранговую аппроксимацию $\mathbf{K}$ произведением матриц малого ранга, полученных с применением случайного проецирования или сэмплирования $\mathbf{K}$, а также псевдообращения (разновидности метода Нистрема, подразд. 6.1 и [115–118]). Кроме того, для ядер, которые являются функциями величин расстояний/сходств векторов большой размерности, быстрая оценка этих расстояний/сходств по предназначенным для этого скетчам/вложениям (см. разд. 2–5) ускоряет оценку ядра. Еще одним подходом является получение таких векторных представлений исходных объектов x, y (возможно, невекторных), скалярное произведение которых позволяет ускорить вычисление $\kappa(x, y)$ точно или приближенно. При таком подходе можно использовать алгоритмы, непосредственно работающие с векторами, что зачастую оказывается более эффективным, чем применение ядерных алгоритмов.

Явное формирование векторов $\boldsymbol{\varphi}(x)$ позволяет непосредственно их использовать. Примером является полиномиальное ядро (8) и явные векторные представления для ядер графов [97, 112, 114, 119–121]. Возможно снижение размерности $\boldsymbol{\varphi}(x)$ методами, описанными в разд. 2–5. Недостатками являются часто очень большая размерность H (например, D^s для полиномиальных векторных ядер (8)) или бесконечная (например, для ядер RBF (9)) или сложность (невозможность) преобразования $\boldsymbol{\varphi}(x)$, а также затраты на снижение размерности.

Рассмотрим методы непосредственного формирования векторных представлений для быстрой оценки ядерных сходств по представлениям или сходствам исходных объектов.

Метод Нистрема (см. подразд. 6.1) требует ОО (адаптация к данным) и формирует векторные представления, сохраняющие сходство, но использует вычислительно сложное разложение по собственным (или сингулярным) значениям.

После публикации [122] получил распространение подход к забывчивому формированию векторных представлений для аппроксимации ядер, для которых известно представление в виде [123]

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

где $\mathbf{w}$ — случайный вектор параметров из некоторого распределения, зависящего от κ, но не от x, y; $\psi(x, \mathbf{w})$ — случайное признаковое отображение (random feature map, RFM) для ядра κ.

Для аппроксимации $\kappa(x, y)$ из распределения выбираются $\mathbf{w}_i$, $i \in [d]$, и вычисляются $\psi_i = \psi(x, \mathbf{w}_i)$, $i \in [d]$, которые присваиваются компонентам вектора $\boldsymbol{\psi}$. Оценка значения ядра определяется как $\kappa^*(x, y) = \langle \boldsymbol{\psi}(x), \boldsymbol{\psi}(y) \rangle / d$. Увеличение d уменьшает дисперсию оценки.

Для инвариантных к сдвигу ядер $\kappa(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{x} - \mathbf{y})$ (гауссова RBF, лапласова, Коши и др.) согласно теореме Бохнера [122] имеется представление вида (10).

Компонент ψ_i формируется нелинейным преобразованием значения $\langle \mathbf{x}, \mathbf{w} \rangle$, где $\mathbf{w} \sim p(\mathbf{w})$, $p(\mathbf{w})$ — обратное преобразование Фурье ядра κ. Например, для RBF (9) $\psi(x, \mathbf{w}, U) = \sqrt{2}\cos(\langle \mathbf{x}, \mathbf{w} \rangle + U)$, где $\mathbf{w}$ — из гауссова распределения $\mathrm{Norm}(\mathbf{0}, \mathbf{I}/\sigma^2)$, $U \sim \mathrm{Unif}[0, 2\pi]$.

Отметим, что, хотя использование векторных представлений RFM показывает хорошие результаты в задачах обучения линейных моделей, если имеется перепад в спектре собственных значений ядра, то векторные представления, полученные методом Нистрема, дают результаты выше, чем RFM [124].

Варианты преобразования приведены в [125]. Аддитивное искажение аппроксимации инвариантных к сдвигу ядер исследовано в [122, 125, 126] (но см. [127]). Для снижения d при том же искажении генерирование $\mathbf{w}$ осуществляют методом квази- Монте-Карло, а также с обучением [128].

Для ускорения случайного проецирования при формировании RFM используют и анализируют [129] наборы из d/D матричных конвейеров $\mathbf{D}_S \mathbf{H} \mathbf{D}_G \mathbf{P} \mathbf{H} \mathbf{D}_R$ с матрицами: $\mathbf{P}$ — случайной перестановки, $\mathbf{D}_G$ — диагональной гауссовой i.i.d., $\mathbf{D}_S$ — диагональной масштабирующей. Матрицу $\mathbf{C}\mathbf{D}_R$ используют в [130]. Формирование векторов $\mathbf{w}$ обучением для более точной аппроксимации заданного ядра и повышение качества классификации при минимальном d исследовано в [131, 132].

Теорема Бохнера также применима для семейства аддитивных гомогенных ядер [133], которые являются функцией скалярной сигнатуры ядра и включают ядра пересечения $\min\{x, y\}$, Хеллингера, χ^2, Иенсена–Шеннона (JS) (подразд. 8.1). В отличие от [122] предлагается вычислять компоненты вектора признаков без случайного сэмплирования с использованием явных аналитических выражений.

Для ядер $\kappa(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{x} + \mathbf{y})$ с $\mathbf{x}, \mathbf{y} \geq 0$, используя расширение теоремы Бохнера, $p(\mathbf{w})$ получают обратным преобразованием Лапласа, $\psi_i(\mathbf{x}) = \exp(-\langle \mathbf{x}, \mathbf{w}_i \rangle)$ [134].

Функция $\kappa(\mathbf{x}, \mathbf{y}) = \mathrm{f}(\langle \mathbf{x}, \mathbf{y} \rangle)$ (например, (8)) является PSD ядром, если $\kappa(\mathbf{x}, \mathbf{y}) = \mathrm{f}(z)$ разложима в ряд Маклорена $\mathrm{f}(z) = \sum_{i=1}^{\infty} a_i z^i$ с $a_i \geq 0$. В [135] предложен скетч $\psi_i(\mathbf{x}) = (a_N 2^{N+1})^{1/2} \prod_{j=1}^{N} \langle \mathbf{x}, \mathbf{w}_j \rangle$, где $\mathbf{w}_j = \mathrm{diag}(\mathbf{D}_R)$, N — случайное число, $\Pr[N = n] = 1/2^{n+1}$. В [136] используют тот факт, что тензорное произведение вектора с самим собой, повторенное s раз, дает вложение в H, соответствующее ядру $\langle \mathbf{x}, \mathbf{y} \rangle^s$. Для каждого вектора создают s разных скетчей размерности d, используя hashing trick [61]. Итоговый скетч размерности d получают вычислением (d-мерного вектора) FFT каждого скетча, их покомпонентным перемножением и выполнением FFT^{-1}. Время получения скетча $O(sD + sd \log d)$, обычно $d = O(D)$.

Для более компактных векторов с лучшей аппроксимацией ядра в [137] вначале получают векторы большой размерности с помощью преобразований [135] или [136], а затем применяют (F)JLT (см. подразд. 3.2). Теорема Бохнера непосредственно не применима к полиномиальному ядру [138], однако для единичных векторов [138] удалось аппроксимировать $p(\mathbf{w})$. Это дает более точную аппроксимацию ядер для больших s.

Методы формирования бинарных векторов для аппроксимации ядер приведены в [1].

8. ДРУГИЕ НАПРАВЛЕНИЯ ИССЛЕДОВАНИЙ

8.1. Вложения расстояний между распределениями. Статистические (вероятностные, информационные) расстояния вводятся для векторов с $x_i \geq 0$ и $\sum_{i=1}^{D} x_i = 1$. Такие векторы можно рассматривать как распределения или точки на D-мерном симплексе — многомерном обобщении треугольника. Многие статистические расстояния не являются метриками и даже несимметричны. Некоторые из них называют (статистическими) дивергенциями.

В [139] отмечают, что для метрических расстояний — статистического Хеллингера $\mathrm{dist}^2_{\mathrm{Hell}} = 1/2\,||\,\mathbf{x}^{1/2} - \mathbf{y}^{1/2}||^2_2$ и Махалонобиса $\mathrm{dist}^2_{\mathrm{Maha}} = (\mathbf{x}-\mathbf{y})^{\mathrm{T}}\,\mathbf{A}(\mathbf{x}-\mathbf{y}) = ||\,\mathbf{L}(\mathbf{x}-\mathbf{y})\,||^2_2$ — возможно снижение размерности с искажением $1\pm\varepsilon$ согласно лемме JL, так как в них используется квадрат евклидова расстояния между преобразованными векторами $\mathbf{x}$, $\mathbf{y}$. Показано, что для вложений в метрические пространства неметрических расстояний (дивергенций) Бхаттачарья $\mathrm{dist}_{\mathrm{Bhat}} = -\ln(\langle \mathbf{x}, \mathbf{y}\rangle)^{1/2}$ и Кульбака–Лейблера $\mathrm{dist}_{\mathrm{KL}} = \sum_{i=1}^{D} x_i \ln x_i / y_i$ существуют конфигурации векторов с произвольно большим мультипликативным искажением A. Для $\mathrm{dist}_{\mathrm{Bhat}}$ выполняется аддитивный аналог леммы JL с искажением $\pm\varepsilon_a(\gamma)$ при $x_i,\ y_i \ge \gamma / D$, $i \in [D]$. Анализ основан на исследовании искажения при оценке одного расстояния по другому, для которого известно снижение размерности по лемме JL ($\mathrm{dist}_{\mathrm{Bhat}} \to \mathrm{dist}_{\mathrm{Hell}}$, $\mathrm{dist}_{\mathrm{KL}} \to (L_2)^2$) без изменения векторных представлений. Отметим, что в рассмотренных вложениях векторы в целевом пространстве не лежат на симплексе.

В [133] приведены явные представления $\boldsymbol{\varphi}(\mathbf{x})$ в H, скалярное произведение которых дает значения ядер JS, Хеллингера, χ^2, а также векторные представления конечной размерности для их аппроксимации. Это позволяет вычислить соответствующие дивергенции $\mathrm{dist}^2_{\mathrm{f}} = ||\,\boldsymbol{\varphi}(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{y})\,||^2$ и их аппроксимации, однако в [133] этот вопрос не исследован.

В [140] для $\mathrm{dist}_{\mathrm{Hell}}$ показано вложение из D-мерного симплекса в d-мерный с искажением $1\pm\varepsilon$ с помощью обычного случайного проецирования, однако векторы должны находиться в определенной области D-симплекса, которая уменьшается с ростом D. В [141] для расстояний JS, Хеллингера, χ^2 ($\mathrm{dist}_{\chi^2} = \sum_{i=1}^{D} (x_i - y_i)^2 / (x_i + y_i)$) и других f-дивергенций определенного в [141] класса показано существование аналога леммы JL с искажением $1\pm\varepsilon$ для любых N точек симплекса. Вначале выполняют нелинейное рандомизированное вложение в $(L_2)^2$ с искажением $1+\varepsilon$ аналогично технике RFM (см. разд. 7) [122]. Затем лемма JL используется для снижения размерности до $d = O(\varepsilon^{-2} \log N)$. Полученные векторы изометрически отображают во внутреннюю область симплекса и далее масштабированием и центрированием относительно центроида симплекса получают итоговые точки на симплексе. Последнее преобразование возможно для f-дивергенций определенного класса [141]. Даны также другие результаты вложений информационных расстояний.

Результаты по оценке информационных расстояний в потоковых моделях приведены в [141, 142].

8.2. Об эквивалентности скетчей и вложений. Как отмечалось в разд. 4, для L_1 (и для L_s, $0 < s \le 2$) существуют эффективные скетчи (но не вложения). Такие скетчи можно использовать для исходных объектов с различными представлениями и мерами расстояния/сходства, вкладываемыми в L_s (при незначительном увеличении искажения).

Однако для вложения многих специализированных метрик в L_s (см. подразд. 6.2) при фиксированной размерности выходных векторов d искажение растет с увеличением эффективной размерности представлений входных объектов со специализированными метриками. Так как скетчи не требуют оценки сходства по метрике (см. подразд. 1.3), возникает вопрос, нельзя ли для таких исходных представлений (объектов) создать скетчи с постоянным искажением непосредственно, минуя промежуточное вложение в L_s. В [75] показано, что это невозможно для

исходных представлений из нормированных векторных пространств (не обязательно L_s) в задаче distance threshold estimation: определить по скетчам, являются объекты близкими или далекими. Представляет интерес получение подобных результатов для более широкого класса метрик (из ненормированных пространств, таких как $\mathrm{dist}_{\mathrm{edit}}$).

8.3. Быстрый поиск по сходству. Линейный поиск по сходству с использованием быстрой оценки расстояния/сходства между объектом-запросом и всеми объектами базы позволяет уменьшить время линейного поиска по исходным мерам сходства, хотя и не дает строгих гарантий качества результатов.

Для поиска по сходству точная оценка расстояний во всем их диапазоне и между всеми объектами избыточна. Достаточно правильно оценивать соотношение расстояний (больше или меньше). Кроме того, высокая точность нужна только для малых расстояний. Это потенциально позволяет использовать не только забывчивые вложения и скетчи, разработанные для быстрой и точной оценки сходств и расстояний (например, с выполнением вариантов лемм JL), но и другие, а также меньшей размерности. Формализация забывчивых вложений для поиска приближенного ближайшего соседа и их пример для расстояния L_2 и данных малой внутренней (intrinsic) размерности приведены в [5].

Как отмечалось в разд. 4, для L_1 не существует вложений с выполнением леммы JL. Однако для линейных вложений случайной i.i.d. матрицей Коши имеется «односторонний» аналог леммы JL, что обеспечивает поиск приближенных ближайших соседей для L_1 по таким вложениям с $d << D$ [81]. Нелинейные вложения в L_1 для малых расстояний приведены в [80] и в подразд. 4.1.

Быстрая оценка расстояний также ускоряет поиск по сходству с использованием имеющихся алгоритмов (индексных структур), работающих на основе вычисления расстояний [143–145]. Хотя вследствие неточности оценок поиск в общем случае приближенный, для сжимающих оценок (см. подразд. 6.1) можно получить точные результаты поиска. Кроме того, ускорение поиска по сходству возможно за счет использования алгоритмов и структур, специализированных для получаемых вещественных векторов (малой и умеренной размерности) с их мерами расстояния/сходства, например, на основе деревьев [145, 146] или локально-чувствительного хэширования LSH [9, 98, 147].

9. ОБСУЖДЕНИЕ

Подытожим преимущества и недостатки основных рассмотренных в обзоре методов формирования вещественных векторных представлений для оценки мер расстояния/сходства и сравним их с методами с обучением.

9.1. Преимущества и недостатки вложений векторов евклидового пространства случайным проецированием. К преимуществам вложений случайным проецированием векторов евклидового пространства (см. разд. 2, 3) (т.е. вещественных векторов, для которых определено и может оцениваться евклидово расстояние, скалярное произведение, угол) относятся: малое искажение оценок при малой размерности итоговых векторов; линейность получения; учет всех компонентов входного вектора и пригодность для любых исходных векторов (неразреженных и разреженных, вещественных и бинарных, с «тяжелыми хвостами»); возможность потоковой обработки с линейной моделью; развитый аппарат анализа в терминах средней ошибки и для худшего случая (варианты леммы JL).

Полученные вещественные векторы (малой размерности) в ряде случаев можно непосредственно использовать в некоторых индексных структурах быстрого поиска по сходству, в векторных методах классификации, аппроксимации и других, линейных и нелинейных, а также для последующего квантования компонентов [1, 147, 148].

Недостатками являются: необходимость формирования случайных матриц; сложность умножения на матрицу (но см. ускорение проецирования в разд. 3);

неприменимость для моделей потоковой обработки с произвольным взвешиванием; невозможность оценить расстояние между подмножеством компонентов исходных векторов; неразреженность (для любых исходных векторов, в том числе разреженных).

9.2. Преимущества и недостатки скетчей, полученных устойчивыми случайными проекциями. Преимущества скетчей для оценки расстояний L_s ($0< s \le 2$), получаемых устойчивым случайным i.i.d. проецированием (см. разд. 4), аналогичны приведенным в подразд. 9.1. Отметим, что для ряда невекторных представлений объектов существует вложение в L_1, поэтому по скетчам для L_1 можно оценить расстояния между исходными представлениями (см. подразд. 6.2).

Недостатки, в дополнение к приведенным в подразд. 9.1, включают: необходимость формирования различных случайных матриц для каждого значения s и сложность генерации случайных чисел из устойчивых распределений; нелинейность оценок; недостаточное исследование ускорения случайного проецирования (но см. разреженные случайные матрицы в [149] из s-Парето распределения); невозможность «автоматического» применения в ряде методов, непосредственно оперирующих векторами.

9.3. Преимущества и недостатки скетчей, полученных сэмплированием. К преимуществам скетчей, полученных равномерным случайным сэмплированием без замещения (см. подразд. 5.1, 5.2) относятся: пригодность одного и того же скетча для оценки любых линейных суммирующих статистик; простота получения скетча; возможность более точных оценок для разреженных векторов; применимость к любым моделям потоковой обработки, включая модели с произвольным взвешиванием компонентов (исходных векторов); возможность работы с выделенными подмножествами компонентов.

К недостаткам относятся: невысокая точность оценок для данных с тяжелыми хвостами и неразреженных; в большинстве случаев сложность анализа ошибки оценки и отсутствие гарантий для худшего случая.

Имеются проблемы с непосредственным применением CRS-скетчей (см. подразд. 5.2) в векторных алгоритмах и LSH. Компоненты различных скетчей с одинаковым номером в скетче не соответствуют один другому, поэтому, например, для обучения линейной модели требуется их разворачивание в векторы исходной размерности. Также отмечают [62] проблемы с построением по ним матрицы ядерных сходств.

Для векторов с тяжелыми хвостами методы взвешенного сэмплирования позволяют повысить точность оценок, однако работают с неотрицательными входными векторами, требуют разработки оценок для различных сходств и расстояний, оценки и расчет их ошибки нетривиальны. Если не имеется тяжелых хвостов, результаты простого сэмплирования могут быть лучше [84].

9.4. Методы оценки сходства с обучением. Большинство рассмотренных в обзоре методов формирования векторных представлений для быстрой оценки расстояний/сходств не учитывает особенностей данных конкретной базы. Адаптация к данным открывает возможности улучшения результатов в применениях быстрой оценки расстояний/сходств. Например, в поиске по сходству можно получить ускорение за счет формирования более компактных представлений, а также повысить качество поиска за счет подстройки к базе представлений и используемых мер расстояния/сходства.

Снижение размерности векторных представлений с использованием обучения как без учителя, так и с учителем, осуществляется линейными и нелинейными методами [150, 151]. Примером линейного (сжимающего) преобразования, формируемого обучением без учителя, является метод главных компонент PCA. Направления проецирования определяются посредством разложения по сингулярным значениям матрицы данных. При снижении размерности PCA обеспечи-

вает наименьшую для линейных методов среднеквадратичную ошибку оценки евклидовых расстояний между векторами обучающей выборки. Однако расстояние между конкретной парой векторов может иметь произвольное искажение (не имеется гарантий худшего случая). Кроме того, теоретически не рассчитывается размерность вложения, обеспечивающая заданное искажение. Методы с обучением используют также для формирования компактных бинарных векторных представлений, отражающих сходство входных объектов [147, 148].

Для повышения качества поиска по сходству используют обучение метрике (metric learning) [152, 153]. Информация о том, какие объекты считать сходными, а какие несходными, задается учителем и используется для настройки параметров мер расстояний/сходств. Так, например, по обучающей выборке настраивается матрица параметров **A** расстояния Махалонобиса (см. подразд. 8.1).

Общим недостатком методов с обучением является их высокая вычислительная сложность. Для некоторых методов нетривиально формирование векторных представлений новых объектов, не используемых в обучении (но см. метод Нистрема, подразд. 6.1). Методы снижения размерности с обучением не всегда решают задачу сохранения исходных расстояний/сходств, поэтому возможны их большие искажения без гарантий сохранения или сокращения расстояний. Кроме того, адаптация к данным предполагает, что данные обучающей выборки и новые данные будут иметь одинаковое распределение, что не всегда выполняется на практике.

Автор благодарен канд. техн. наук А.М. Соколову за обсуждения.

СПИСОК ЛИТЕРАТУРЫ

1. Рачковский Д.А. Бинарные векторы для быстрой оценки расстояний и сходств // Кибернетика и системный анализ. — 2017. — **53**, N 1 (в печати).
2. Deza M., Deza E. Encyclopedia of distances. — Berlin; Heidelberg: Springer, 2016. — 756 p.
3. Indyk P., Matousek J. Low-distortion embeddings of finite metric spaces // Handbook of discrete and computational geometry. — Boca Raton (FL): Chapman & Hall/CRC, 2004. — P. 177–196.
4. Matousek J. Lecture notes on metric embeddings. — 2013. — 126 p.
5. Indyk P., Naor A. Nearest-neighbor-preserving embeddings // ACM Trans. Algorithms. — 2007. — **3**, N 3. — Article No 31. —
6. Batu T., Ergun F., Sahinalp C. Oblivious string embeddings and edit distance approximations // SODA'06. — 2006. — P. 792–801.
7. Cormode G., Garofalakis M., Haas P.J., Jermaine C. Synopses for massive data: Samples, histograms, wavelets, sketches // Foundations and Trends® in Databases. — 2012. — **4**, N 1–3. — P. 1–294.
8. Johnson W.B., Lindenstrauss J. Extensions of Lipshitz mapping into Hilbert space // Contemporary Mathematics. — 1984. — **26.** — P. 189–206.
9. Indyk P., Motwani R. Approximate nearest neighbors: Towards removing the curse of dimensionality // Proc. 30th ACM Symp Theory of Computing. — 1998. — P. 604–613.
10. Achlioptas D. Database-friendly random projections: Johnson–Lindenstrauss with binary coins // Journal of Computer and System Sciences. — 2003. — **66**, N 4. — P. 671–687.
11. Matousek J. On variants of the Johnson Lindenstrauss lemma // Random Structures and Algorithms. — 2008. — **33**, N 2. — P. 142–156.
12. Jayram T.S., Woodruff D.P. Optimal bounds for Johnson–Lindenstrauss transforms and streaming problems with subconstant error // ACM Trans. on Algorithms. — 2013. — **9**, N 3. — Article 26.
13. Kane D.M., Meka R., Nelson J. Almost optimal explicit Johnson–Lindenstrauss families // Proc. RANDOM'11. — 2011. — P. 628–639.
14. Larsen K. G., Nelson J. The Johnson–Lindenstrauss lemma is optimal for linear dimensionality reduction // Proc. ICALP'16. — 2016. —
15. Alon N. Problems and results in extremal combinatorics. I // Discrete Mathematics. — 2003. — **273**, N 1–3. — P. 31–53.

16. Vershynin R. Introduction to the non-asymptotic analysis of random matrices // Compressed Sensing, Theory and Applications. — 2012. — P. 210–268.
17. Buldygin V., Moskvichova K. The sub-gaussian norm of a binary random variable // Theory of Probability and Mathematical Statistics. — 2013. — **86.** — P. 33–49.
18. Dirksen S. Dimensionality reduction with subgaussian matrices: a unified theory // Foundations of Computational Mathematics. — 2015. — P. 1–30.
19. Baraniuk R. G., Davenport M., DeVore R. A., Wakin M. A simple proof of the restricted isometry property for random matrices // Constr. Approx. — 2008. — **28**, N 3. — P. 253–263.
20. Nelson J., Price E., Wootters M. New constructions of RIP matrices with fast multiplication and fewer rows // Proc. SODA'14. — 2014. — P. 1515–1528.
21. Krahmer F., Ward R. New and improved Johnson–Lindenstrauss embeddings via the Restricted Isometry Property // SIAM J. Math. Anal. — 2011. — **43**, N 3. — P. 1269–1281.
22. Ailon N., Rauhut H. Fast and rip-optimal transforms // Discrete and Computational Geometry. — 2014. — **52**, N 4. — P. 780–798.
23. Haviv I., Regev O. The restricted isometry property of subsampled Fourier matrices // Proc. SODA'16. — 2016. — P. 288–297.
24. Bourgain J., Dirksen S., Nelson J. Toward a unified theory of sparse dimensionality reduction in Euclidean space // Geometric and Functional Analysis. — 2015. — **25**, N 4. — P. 1009–1088.
25. Gordon Y. On Milman's inequality and random subspaces which escape through a mesh in R^n // Geometric Aspects of Functional Analysis. — 1988. — P. 84–106.
26. Schechtman G. Two observations regarding embedding subsets of Euclidean spaces in normed spaces // Adv. Math. — 2006. — **200**, N 1. — P. 125–135.
27. Oymak S., Recht B., Soltanolkotabi M. Isometric sketching of any set via the restricted isometry property // arXiv:1506.03521. — 6 Oct 2015.
28. Klartag B., Mendelson S. Empirical processes and random projections // Journal of Functional Analysis. — 2005. — **225**, N 1. — P. 229–245.
29. Karnin Z, Rabani Y, Shpilka A. Explicit dimension reduction and its applications // SIAM J. Comput. — 2012. — **41**, N 1. — P. 219–249.
30. Kane D.M., Nelson J. Sparser Johnson-Lindenstrauss transforms // Journal of the ACM. — 2014. — **61**, N 1. — P. 4:1–4:23.
31. Engebretsen L., Indyk P., O'Donnell R. Derandomized dimensionality reduction with applications // Proc. SODA'02. — 2002. — P. 705–712.
32. Sivakumar D. Algorithmic derandomization via complexity theory // Proc. 34th Annual ACM Symposium on Theory of Computing, Montreal, QC, 2002, ACM, New York, — 2002. — P. 619–626.
33. Li P., Hastie T.J., Church K.W. Very sparse random projections // Proc. KDD'06. — 2006. — P. 287–296.
34. Rachkovskij D.A. Vector data transformation using random binary matrices // Cybernetics and Systems Analysis — 2014 — **50**, N 6. — P. 960–968.
35. Vempala S.S. The random projection method. — American Math. Soc., 2004. — 105 p.
36. Arriaga R.I., Vempala S. An algorithmic theory of learning: Robust concepts and random projection // Machine Learning. — 2006. — **63**, N 2. — P.161–182.
37. Kabán A. Improved bounds on the dot product under random projection and random sign projection // Proc. KDD'15. — 2015. — P. 487–496.
38. Yi X., Caramanis C., Price E. Binary embedding: Fundamental limits and fast algorithm // arXiv:1502. 05746. — 19 Feb 2015.
39. Magen A. Dimensionality reductions in $\ell 2$ that preserve volumes and distance to affine spaces // Discrete Comput. Geom. — 2007. — **38**, N 1. — P. 139–153.
40. Plan Y., Vershynin R. One-bit compressed sensing by linear programming // Communications on Pure and Applied Mathematics. — 2013. — **66**, N 8. — P. 1275–1297.
41. Ailon N., Chazelle B. The Fast Johnson–Lindenstrauss Transform and approximate nearest neighbors // SIAM J. Comput. — 2009. — **39**, N 1. — P. 302–322.
42. Plan Y., Vershynin R. Dimension reduction by random hyperplane tessellations // Discrete and Computational Geometry. — 2014. — **51**, N 2. — P. 438-461.

43. Liberty E., Zucker S.W. The mailman algorithm: A note on matrix-vector multiplication // Inf. Process. Lett. — 2009. — **109**, N 3. — P. 179–182.

44. Rachkovskij D., Slipchenko S. Similarity-based retrieval with structure-sensitive sparse binary distributed representations // Computational Intelligence. — 2012. — **28**, N 1. — P. 106–129.

45. Kanerva P., Kristoferson J., Holst A. Random indexing of text samples for latent semantic analysis // 22nd Annual Conference of the Cognitive Science Society. — 2000. — P. 1036.

46. Мисуно И.С., Рачковский Д.А., Слипченко С.В. Векторные и распределенные представления, отражающие меру семантической связи слов // Математические машины и системы. — 2005. — № 3. — С. 50–67.

47. Rachkovskij D.A., Misuno I.S., Slipchenko S.V. Randomized projective methods for construction of binary sparse vector representations // Cybernetics and Systems Analysis. — 2012. — **48**, N 1. — P. 146–156.

48. Rachkovskij D.A. Formation of similarity-reflecting binary vectors with random binary projections // Cybernetics and Systems Analysis. — 2015. — **51**, N 2. — P. 313–323.

49. Ailon N., Liberty E. Fast dimension reduction using rademacher series on dual BCH codes // Discrete and Computational Geometry. — 2009. — **42**, N 4. — P. 615–630.

50. Liberty E., Ailon N., Singer A. Dense fast random projections and lean walsh transforms // Discrete and Computational Geometry. — 2011. — **45**, N 1. — P. 34–44.

51. Ailon N., Liberty E. An almost optimal unrestricted fast Johnson–Lindenstrauss transform // ACM Transactions on Algorithms. — 2013. — **9**, N 3. — Article No 21.

52. Rauhut H., Romberg J., Tropp J. Restricted isometries for partial random circulant matrices // Applied and Computational Harmonic Analysis. — 2012. — **32**, N 2. — P. 242–254.

53. Hinrichs A., Vybiral J. Johnson-lindenstrauss lemma for circulant matrices // Random Structures & Algorithms. — 2011. — **39**, N 3. — P. 391–398.

54. Vybiral J. A variant of the Johnson–Lindenstrauss lemma for circulant matrices // Journal of Functional Analysis. — 2011. — **260**, N 4. — P. 1096–1105.

55. Krahmer F., Mendelson S., Rauhut H. Suprema of chaos processes and the restricted isometry property // Comm. Pure Appl. Math. — 2014. — **67**, N 11. — P. 1877–1904.

56. Zhang H., Cheng L. New bounds for circulant Johnson–Lindenstrauss embeddings // Communications in Mathematical Sciences. — 2014. — **12**, N 4. — P. 695–705.

57. Yang Z., Moczulski M., Denil M., de Freitas N., Smola A., Song L., Wang. Z. Deep fried convnets // Proc. ICCV'15. — 2015. — P. 1476–1483.

58. Cheng Y., Yu F.X., Feris R.S., Kumar S., Choudhary A., Chang S.- F. An exploration of parameter redundancy in deep networks with circulant projections // Proc. ICCV'15. — 2015. — P. 2857–2865.

59. Sindhwani V., Sainath T., Kumar S. Structured transforms for smallfootprint deep learning // Proc. NIPS'15. — 2015. — P. 3070–3078.

60. Moczulski M., Denil M., Appleyard J., de Freitas N. Acdc: A structured efficient linear layer // ICLR'16. — 2016. — arXiv:1511.05946.

61. Weinberger K., Dasgupta A., Langford J., Smola A., Attenberg J. Feature hashing for large scale multitask learning // Proc. ICML'09. — 2009. — P. 1113–1120.

62. Li P., Shrivastava A., Moore J.L., König A.C. Hashing algorithms for large-scale learning // Proc. NIPS'11. — 2011. — P. 2672–2680.

63. Dasgupta A., Kumar R., Sarlos T. A sparse johnson-lindenstrauss transform // Proc. STOC'10. — 2010. — P. 341–350.

64. Kane D.M., Nelson J. A derandomized sparse JohnsonLindenstrauss transform // Electronic Colloquium on Computational Complexity. — 2010. — **17.** — Article 98.

65. Braverman V, Ostrovsky R, Rabani Y. Rademacher chaos, random Eulerian graphs and the sparse Johnson–Lindenstrauss transform // arXiv:1011.2590. — 11 Nov. 2010.

66. Nelson J., Nguyen H.L. Sparsity lower bounds for dimensionality reducing maps // Proc. STOC'13. — 2013. — P. 101–110.

67. Ventkatasubramanian S., Wang Q. The johnson-lindenstauss transform: An empirical study // Proc. ALENEX'11. — 2011. — P. 164–173.

68. Lee J.R., Naor A. Embedding the diamond graph in Lp and dimension reduction in L1 // Geometric and Functional Analysis. — 2004. — **14**, N 4. — P. 745–747.
69. Brinkman B., Charikar M. On the impossibility of dimension reduction in l1 // Journal of the ACM. — 2005. — **52**, N 5. — P. 766–788.
70. Lee J., Mendel M., Naor A. Metric structures in l1: Dimension, snowflakes, and average distortion // European Journal of Combinatorics. — 2005. — **26**, N 8. — P. 1180–1190.
71. Andoni A., Charikar M., Neiman O., Nguyen H.L. Near linear lower bounds for dimension reduction in L1 // Proc. FOCS'11. — 2011. — P. 315–323.
72. Newman I., Rabinovich Y. Finite volume spaces and sparsification // arXiv:1002.3541. — 2 Aug. 2010.
73. Figiel T., Lindenstrauss J., Milman V.D. The dimension of almost spherical sections of convex bodies // Acta Math. — 1977. — **139**, N 1. — P. 53–94.
74. Johnson W.B., Schechtman G. Embedding l_p^m into l_1^n // Acta Math. — 1982. — **149**, N 1. — P. 71–85.
75. Andoni A., Krauthgamer R., Razenshteyn I.P. Sketching and embedding are equivalent for norms // Proc. STOC'15. — 2015. — P. 479–488.
76. Bar-Yossef Z., Jayram T. S., Kumar R., Sivakumar D. An information statistics approach to data stream and communication complexity // J. Comput. Syst. Sci. — 2004. — **68**, N 4. — P. 702–732.
77. Berinde R., Gilbert A. C., Indyk P., Karloff H., Strauss M.J. Combining geometry and combinatorics: A unified approach to sparse signal recovery // AAC on CCC'08. — 2008. — P. 798–805.
78. Krahmer F., Ward R. A unified framework for linear dimensionality reduction in L1 // Results in Mathematics. — 2016. — **70**, N 1. — P. 209–231.
79. Allen-Zhu Z., Gelashvili R., Razenshteyn I. Restricted isometry property for general p-norms // Proc. SoCG'15. — 2015. — P. 451–460.
80. Bartal Y, Gottlieb L.-A. Dimension reduction techniques for ℓp ($1 \le p \le 2$), with applications // Proc. SoCG'16. — 2016. — P. 16:1–16:15.
81. Indyk P. Stable distributions, pseudorandom generators, embeddings, and data stream computation // Journal of the ACM. — 2006. — **53**, N 3. — P. 307–323.
82. Li P. Estimators and tail bounds for dimension reduction in $\ell\alpha$ ($0 < \alpha \le 2$) using stable random projections // Proc. SODA'08. — 2008. — P. 10–19.
83. Li P. Computationally efficient estimators for dimension reductions using stable random projections // Proc. ICDM'08. — 2008. — P. 403–412.
84. Duffield N., Lund C., Thorup M. Priority sampling for estimating arbitrary subset sums // J. Assoc. Comput. Mach. — 2007. — **54**, N 6. — Article No 32.
85. Cohen E. Distance queries from sampled data: Accurate and efficient // KDD'14. — 2014. — P. 681–690.
86. Li P., Church K.W, Hastie T.J. One sketch for all: Theory and applications of conditional random sampling // Proc. NIPS'08. — 2008. — P. 953–960.
87. Thorup M. Bottom-k and priority sampling, set similarity and subset sums with minimal independence // Proc. STOC'13. — 2013. — P. 371–378.
88. Charikar M. Similarity estimation techniques from rounding algorithms // Proc. STOC'02. — 2002. — P. 380–388.
89. Ioffe S. Improved consistent sampling, weighted minhash and L1 sketching // Proc. ICDM'10. — 2010. — P. 246–255.
90. Li P. Generalized min-max kernel and generalized consistent weighted sampling // arXiv:1605.05721. — 23 May 2016.
91. Cohen E. Estimation for monotone sampling: competitiveness and customization // Proc. PODC'14. — 2014. — P. 124–133.
92. Williams C.K.I., Seeger M. Using the Nystrom method to speed up kernel machines // Proc. NIPS'00. — 2000. — P. 682–688.
93. Hjaltason G.R., Samet H. Properties of embedding methods for similarity searching in metric spaces // IEEE Trans. PAMI. — 2003. — **25**, N 5. — P. 530–549.

94. Platt J.C. FastMap, MetricMap, and Landmark MDS are all Nystrom algorithms // Proc. AISTATS'05. — 2005. — P. 261–268.
95. Pekalska E., Duin R.P.W. The dissimilarity representation for pattern recognition, Foundations and applications. — Singapore: World Scientific, 2005. — 607 p.
96. Chavez E., Graff M., Navarro G.,. Tellez E.S. Near neighbor searching with K nearest references // Information Systems. — 2015. — **51**(C). — P. 43–61.
97. Riesen K., Neuhaus M., Bunke H. Graph embedding in vector spaces by means of prototype selection // Proc. GbRPR'07. — 2007. — P. 383–393.
98. Andoni A. Nearest neighbor search: the old, the new, and the impossible: PhD thesis, Massachusetts Institute of Technology. — 2009. — 178 p.
99. Levenshtein V.I. Binary codes capable of correcting deletions, insertions, and reversals // Soviet Physics - Doklady. — 1966. — **10**, N 8. — P. 707–710.
100. Navarro G. A guided tour to approximate string matching // ACM CSUR. — 2001. — **33**, N 1. — P. 31–88.
101. Backurs A., Indyk P. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false) // Proc. STOC'15. — 2015. — P. 51–58.
102. Bringmann K., Kunnemann M. Quadratic conditional lower bounds for string problems and dynamic time warping // Proc. FOCS'15. — 2015. — P. 79–97.
103. Cormode G., Muthukrishnan S. The string edit distance matching problem with moves // ACM Trans. Algorithms. — 2007. — **3**, N 1. — P. 2:1–2:19.
104. Sokolov A. Vector representations for efficient comparison and search for similar strings // Cybernetics and System Analysis. — 2007. — **43**, N 4. — P. 484–498.
105. Ostrovsky R., Rabani Y. Low distortion embeddings for edit distance // Journal of the ACM. — 2007. — **54**, N 5. — P. 23–36.
106. Andoni A., Onak K. Approximating edit distance in near-linear time // SIAM Journal on Computing. — 2012. — **41**, N 6. — P. 1635–1648.
107. Andoni A., Krauthgamer R., Onak K. Polylogarithmic approximation for edit distance and the asymmetric query complexity // Proc. FOCS'10. — 2010. — P. 377–386.
108. Krauthgamer R., Rabani Y. Improved lower bounds for embeddings into L1 // Proc. SODA'06. — 2006. — P. 1010–1017.
109. Andoni A., Goldberger A., McGregor,A., Porat E. Homomorphic fingerprints under misalignments: sketching edit and shift distances // Proc. STOC'13. — 2013. — P. 931–940.
110. Scholkopf B., Smola A.J. Learning with kernels, Support Vector Machines, regularization, optimization, and beyond. — Cambridge: MIT Press, 2001. — 626 p.
111. Vishwanathan S.V.N., Schraudolph N.N., Kondor R., Borgwardt K.M. Graph kernels // Journal of Machine Learning Research. — 2010. — **11.** — P. 1201–1242.
112. Conte D., Ramel J.Y., Sidere N., Luqman M.M., Gauzere B., Gibert J., Brun L., Vento M. A comparison of explicit and implicit graph embedding methods for pattern recognition // LNCS. — 2013. — **7877.** — P. 81–90.
113. Feragen A., Kasenburg N., Petersen J., de Bruijne M., Borgwardt K.M. Scalable kernels for graphs with continuous attributes // Proc. NIPS'13. — 2013. — P. 216–224.
114. Foggia P., Percannella G., Vento M. Graph matching and learning in pattern recognition in the last 10 years // Int. J. Pattern Recog. Artif. Intell. — 2014. — **28**, N 1. — P. 1–40.
115. Halko N., Martinsson P.-G., Tropp J.A.. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions // SIAM Review. — 2011. — **53**, N 2. — P. 217–288.
116. Gittens A., Mahoney M.W. Revisiting the Nystrom method for improved large-scale machine learning // Proc. ICML'13. — 2013. — P. 567–575.
117. Cohen M.B, Lee Y.T., Musco C, Musco C., Peng R., Sidford A. Uniform sampling for matrix approximation // Proc. ITCS'15. — 2015. — P. 181– 190.
118. Wang S. Luo Luo, Zhihua Zhang. SPSD matrix approximation vis column selection: Theories, algorithms, and extensions // Journal of Machine Learning Research. — 2016. — **17.** — P. 1–49.
119. Shervashidze N., Vishwanathan S. V. N., Petri T., Mehlhorn K., Borgwardt K. Efficient graphlet kernels for large graph comparison // JMLR: W&CP. — 2009. — **5.** — P. 488–495.

120. Gibert J., Valveny E., Bunke H. Embedding of graphs with discrete attributes via label frequencies // Int. J. Patt. Recogn. Artif. Intell. — 2013. — **27**, N 3. — P. 1–27.
121. Kriege N., Neumann M., Kersting K., Mutzel P. Explicit versus implicit graph feature maps: A computational phase transition for walk kernels // Proc. ICDM'14. — 2014. — P. 881–886.
122. Rahimi A., Recht B. Random features for large-scale kernel machines // Proc. NIPS'07. — 2007. — P. 1177–1184.
123. Tropp J.A. An introduction to matrix concentration inequalities // Foundations and Trends® in Machine Learning. — 2015. — **8**, N 1–2. — P. 1–230.
124. Yang T., Li Y.-F., Mahdavi M., Jin R., Zhou Z.-H. Nystrom method vs random Fourier features: A theoretical and empirical comparison // Proc. NIPS'12. — 2012. — P. 485– 493.
125. Sutherland D. J., Schneider J. On the error of random Fourier features // Proc. UAI. — 2015. — P. 862–871.
126. Sriperumbudur B. K., Szabo Z. Optimal rates for random Fourier features // Proc. NIPS'15. — 2015. — P. 1144–1152.
127. Chen D., Phillips J.M. Relative error embeddings of the Gaussian kernel distance // arXiv:1602.05350. —
128. Yang J., Sindhwani V., Avron H., Mahoney M.W. Quasi-Monte Carlo feature maps for shift-invariant kernels // Proc. ICML'14. — 2014. — P. 485–493.
129. Le Q., Sarlos T., Smola A.J. Fastfood - Computing Hilbert space expansions in loglinear time // JMLR W&CP. — 2013. — **28**, N 3. — P. 244–252.
130. Feng C., Hu Q., Liao S. Random feature mapping with signed circulant matrix projection // Proc. IJCAI'15. — 2015. — P. 3490–3496.
131. Yu F.X., Kumar S., Rowley H., Chang S.-F. Compact nonlinear maps and circulant extensions // arXiv:1503.03893. — 12 Mar 2015.
132. Choromanski K., Sindhwani V. Recycling randomness with structure for sublinear time kernel expansions // Proc. ICML. — 2016. — P. 2502–2510.
133. Vedaldi A., Zisserman A. Efficient additive kernels via explicit feature maps // IEEE Trans. PAMI. — 2012. — **34**, N 3. — P. 480– 492.
134. Yang J., Sindhwani V., Fan Q., Avron H., Mahoney M.W. Random laplace feature maps for semigroup kernels on histograms // Proc. CVPR'14. — 2014. — P. 971–978.
135. Kar P., Karnick H. Random feature maps for dot product kernels // Proc. ICAIS'12. — 2012. — P. 583–591.
136. Pham N., Pagh R. Fast and scalable polynomial kernels via explicit feature maps // Proc. KDD'13. — 2013. — P. 239–247.
137. Hamid R., Xiao Y., Gittens A., DeCoste D. Compact random feature maps // Proc. ICML'14. — 2014. — P. 19–27.
138. Pennington J., Yu F.X., Kumar S. Spherical random features for polynomial kernels // Proc. NIPS'15. — 2015. — P. 1846–1854.
139. Bhattacharya A., Kar P., Pal M. On low distortion embeddings of statistical distance measures into low dimensional spaces // Proc. DEXA'09. — 2009. — P. 164–172.
140. Kyng R. J., Phillips J. M., Venkatasubramanian S. Johnson–Lindenstrauss dimensionality reduction on the simplex // Proc. FWCG'10. — 2010.
141. Abdullah A., McGregor A., Kumar R., Vassilvitskii S., Venkatasubramanian S. Sketching, embedding, and dimensionality reduction in information spaces // JMLR: W&CP. — 2016. — **41.** — P. 948–956.
142. Guha S., Indyk P., McGregor A. Sketching information divergences // COLT. — 2007. — P. 424–438.
143. Chávez E., Navarro G., Baeza-Yates R., Marroquín J.L. Searching in metric spaces // ACM Computing Surveys. — 2001. — **33**, N 3. — P. 273–321.
144. Zezula P., Amato G., Dohnal V., Batko M. Similarity search: The metric space approach. — New York: Springer, 2006. — 220 p.
145. Samet H. Foundations of multidimensional and metric data structures. — San Francisco: Morgan Kaufmann, 2006. — 1024 p.
146. Muja M., Lowe D.G. Scalable nearest neighbor algorithms for high dimensional data // IEEE Trans. on PAMI. — 2014. — **36**, N 11. — P. 2227–2240.

147. Wang J., Shen H.T., Song J., Ji J. Hashing for similarity search: A survey // arXiv:1408.2927. — 13 Aug 2014.

148. Wang J., Liu W., Kumar S., Chang S.-F. Learning to hash for indexing big data: A survey // arXiv:1509.05472. — 17 Sep 2015.

149. Li P. Very sparse stable random projections for dimension reduction in $l\alpha$ $(0 < \alpha \le 2)$ norm // Proc. SIGKDD'07. — 2007. — P. 440–449.

150. Cunningham J., Ghahramani Z. Linear dimensionality reduction: Survey, insights, and generalizations // Journal of Machine Learning Research. — 2015. — **16.** — P. 2859–2900.

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

152. Kulis B. Metric learning: a survey // Foundations and Trends® in Machine Learning. — 2012. — **5**, N 4. — P. 287–364.

153. Bellet A., Habrard A., Sebban M. A survey on metric learning for feature vectors and structured data // arXiv:1306.6709. — 12 Feb 2014.

Надійшла до редакції 13.05.2016

Д.О. Рачковський

ДІЙСНІ ВКЛАДЕННЯ І СКЕТЧІ ДЛЯ ШВИДКОЇ ОЦІНКИ ВІДСТАНЕЙ ТА СХОЖОСТЕЙ

Анотація. Розглянуто методи і алгоритми швидкої оцінки мір відстані/схожості даних за дійсними векторними представленнями малої розмірності. Досліджено методи без навчання, з використанням випадкової проекції та семпліювання. Вхідні дані є, в основному, векторами великої розмірності з різними мірами відстані (евклідове, манхеттенове, статистичне та ін.) і схожості (скалярний добуток та ін.). Обговорюються також векторні представлення невекторних даних. Отримані вектори можуть також застосовуватися в алгоритмах пошуку за схожістю, машинного навчання тощо.

Ключові слова: відстань, схожість, вкладення, скетчі, зниження розмірності, випадкові проеціювання, семпліювання, лема Джонсона–Лінденштрауса, ядерна схожість, пошук за схожістю.

D.A. Rachkovskij

REAL-VALUED EMBEDDINGS AND SKETCHES FOR FAST DISTANCEAND SIMILARITY ESTIMATION

Abstract. This survey paper focuses on methods and algorithms for fast estimation of data distance/similarity measures. The estimation is done by real-valued vector representations of small dimension. The discussed methods do not use learning and mainly use random projection and sampling. Initial data are mainly high-dimensional vectors with different distance measures (Euclidean, Manhattan, statistical, etc.) and similarities (dot product etc.). Vector representations of non-vector data are discussed as well. The resultant vectors can also be used for similarity search algorithms, machine learning, etc.

Keywords: distance, similarity, embeddings, sketches, dimensionality reduction, random projection, sampling, Johnson–Lindenstrauss lemma, kernel similarity, similarity search.

Рачковский Дмитрий Андреевич,
доктор техн. наук, ведущий научный сотрудник Международного научно-учебного центра информационных технологий и систем НАН и МОН Украины, Киев, e-mail: dar@infrm.kiev.ua.