НОВІ ЗАСОБИ
КІБЕРНЕТИКИ, ІНФОРМАТИКИ,
ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
ТА СИСТЕМНОГО АНАЛІЗУ

Д.А. РАЧКОВСКИЙ

УДК 004.22+004.93′11

БИНАРНЫЕ ВЕКТОРЫ ДЛЯ БЫСТРОЙ ОЦЕНКИ РАССТОЯНИЙ И СХОДСТВ

Аннотация. Рассмотрены методы и алгоритмы быстрой оценки мер расстояния/сходства исходных данных по векторным представлениям с бинарными или целочисленными компонентами, полученным из исходных данных, которые являются в основном векторами большой размерности с различными мерами расстояния (угловое, евклидово и др.) и сходства (косинус угла, скалярное произведение и др.). Обсуждены методы без обучения, использующие главным образом случайное проецирование с последующим квантованием, а также сэмплирование. Полученные векторы можно применять в алгоритмах поиска по сходству, машинного обучения и др.

Ключевые слова: расстояние, сходство, вложения, скетчи, случайное проецирование, сэмплирование, бинаризация, квантование, лемма Джонсона–Линденштраусса, ядерное сходство, поиск по сходству, локально-чувствительное хэширование.

В настоящем обзоре рассмотрены методы и алгоритмы быстрой оценки мер расстояния/сходства исходных данных по векторным представлениям с бинарными или целочисленными компонентами, полученным из исходных данных. Представлены в основном методы без обучения, т.е. без адаптации к данным. (Вещественные векторы для быстрой оценки расстояний и сходств обсуждаются в обзоре [1].)

Функции (меры) расстояния и сходства [2–4] широко применяются для поиска по сходству, анализа данных, машинного обучения (например, кластеризация, классификация, аппроксимация) во многих предметных областях [2]. Если точное вычисление расстояния/сходства исходных представлений объектов требует больших вычислительных затрат, актуальна быстрая приближенная оценка их значений. Для получения такой оценки используют преобразование исходных представлений различного типа (векторных и невекторных) в векторные представления, по которым можно вычислительно проще оценить сходство исходных. Так, высокую точность оценки евклидовых расстояний между исходными вещественными векторами большой размерности дает евклидово расстояние между полученными из них случайным проецированием ([5–8, 1] и разд. 1) вещественными векторами малой размерности. Если оценку исходной меры расстояния удается получить по результирующим векторам с бинарными компонентами (например, разд. 1), то ускорения (и экономии памяти на представление и хранение) можно добиться даже без снижения размерности векторов за счет эффективности представления и обработки бинарных векторов (подразд. 6.1).

Преобразованные представления исходных объектов, по которым оцениваются исходные меры расстояния или сходства, относят к категориям вложений или скетчей ([9, 1] и ссылки к ним). Вложениями обычно называют представления, вычисление расстояний/сходств которых непосредственно дает оценку исходных. Понятие скетча обычно более широкое, чем вложения. Скетчами называют компактные представления исходных объектов, которые применяют для оценки их характеристик, не обязательно расстояний или сходств. В скетчах, предназначенных для расстояний/сходств, получение оценки иногда требует вычисления некоторой (сложной, нелинейной) функции значений компонентов скетчей или даже аналитически не известной, но протабулированной функции. Скетчи используют и для потоковой обработки данных, т.е. когда представления объектов задают последовательностью компонентов или их приращений.

Часто и вложения, и скетчи являются векторными представлениями. В настоящем обзоре рассмотрены векторные вложения и скетчи с бинарными или дискретными (целочисленными) компонентами для оценки расстояний/сходств главным образом исходных векторов большой размерности, вещественных и бинарных. В дальнейшем скетчами будем называть результирующие векторные представления исходных объектов (включая и результаты вложений).

Для многих применений заранее не известен набор объектов, расстояния или сходства которых надо оценивать. Поэтому востребованы забывчивые (oblivious) методы формирования вложений или скетчей, не зависящие от других объектов [10] или, по крайней мере, применимые к новым объектам [11]. Так как забывчивое формирование векторных представлений, обеспечивающих приемлемую точность оценки сходства, является трудной задачей, большинство забывчивых методов рандомизированные (используют псевдослучайные числа и обеспечивают лишь вероятностные оценки расстояний/сходств по полученным вложениям/скетчам).

Ряд векторных представлений с бинарными и целочисленными компонентами, помимо оценки мер расстояния/сходства исходных объектов (и применения в основанных на расстояниях/сходствах алгоритмах), можно также использовать (непосредственно или при надлежащем преобразовании) с арсеналом алгоритмов для векторных данных (линейные и нелинейные методы классификации и аппроксимации, индексные структуры быстрого поиска по сходству (подразд. 6.2), отбор информативных признаков и др.). Так, если скалярное произведение векторов аппроксимирует ядерное сходство (разд. 3) исходных объектов, для больших обучающих выборок более эффективны методы обучения линейных моделей, а не вычислительно сложные ядерные методы. Алгоритмы, специализированные для бинарных векторных данных, часто превосходят по скорости выполнения и использованию памяти (подразд. 6.1) алгоритмы для вещественных векторов.

Структура обзора следующая: в разд. 1 рассмотрены бинаризованные вложения для оценки угла между исходными векторами, которые формируют случайным проецированием и покомпонентной бинаризацией по знаку результата; в разд. 2 описано получение скетчей с дискретными элементами с помощью локально-чувствительного хэширования; в разд. 3 рассмотрены бинарные векторы для аппроксимации ядерных сходств; в разд. 4 представлены скетчи для оценки мер расстояния/сходства исходных бинарных векторов; в разд. 5 приведены бинарные разреженные скетчи (с малым числом ненулевых компонентов) и целочисленные скетчи квантованием случайного проецирования; в разд. 6 рассмотрены эффективность применения бинарных векторов и адаптация к данным.

1. БИНАРИЗОВАННЫЕ ВЛОЖЕНИЯ ДЛЯ ОЦЕНКИ УГЛА МЕЖДУ ВЕКТОРАМИ

Пусть $\mathbf{x}, \mathbf{y}$ — исходные вещественные векторы размерности D, $\mathbf{r}$ — случайный вектор размерности D с компонентами, которые являются независимо и одинаково распределенными (i.i.d.) случайными величинами (с.в.) из гауссова распределения ($\mathbf{r} \sim \mathrm{Norm}(\mathbf{0}, \mathbf{I}_D)$), $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^{D} u_i v_i$ — скалярное произведение векторов $\mathbf{u}$ и $\mathbf{v}$, $\mathrm{sign}(z) = 1$ при $z \geq 0$ и $\mathrm{sign}(z) = -1$ при $z < 0$. Для с.в. $h_{\mathbf{r}}(\mathbf{x}) = \mathrm{sign}(\langle \mathbf{r}, \mathbf{x} \rangle)$ выполняется [12, 13]

$$\Pr\{h(\mathbf{x}) \neq h(\mathbf{y})\} = \arccos(\mathbf{x}, \mathbf{y}) / \pi = \theta(\mathbf{x}, \mathbf{y}) / \pi, \tag{1}$$

где Pr — вероятность, $\arccos(\mathbf{x}, \mathbf{y}) \equiv \theta(\mathbf{x}, \mathbf{y})$ — величина угла между векторами $\mathbf{x}$ и $\mathbf{y}$.

Обозначим $\mathrm{dist}_{\mathrm{Ham}}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{d} 1\{a_i \neq b_i\}$ расстояние Хэмминга между бинарными векторами $\mathbf{a}$, $\mathbf{b}$ размерности d ($1\{\cdot\}$ — индикаторная функция); $\mathrm{dist}_{\mathrm{ham}}(\mathbf{a}, \mathbf{b}) = \mathrm{dist}_{\mathrm{Ham}}(\mathbf{a}, \mathbf{b}) / d$ — нормированное к диапазону $[0, 1]$ расстояние Хэмминга. Оценкой $\Pr^*$ вероятности $\Pr\{h(\mathbf{x}) \neq h(\mathbf{y})\}$ (1) является

$$\Pr^*\{h(\mathbf{x}) \neq h(\mathbf{y})\} = \mathrm{dist}_{\mathrm{ham}}(\mathbf{h}(\mathbf{x}), \mathbf{h}(\mathbf{y})), \tag{2}$$

где $\mathbf{h}(\mathbf{x}) = \mathrm{sign}(\mathbf{R}\mathbf{x})$, $\mathbf{h}(\mathbf{y}) = \mathrm{sign}(\mathbf{R}\mathbf{y})$ с покомпонентной операцией sign, $\mathbf{R}(d \times D)$ — случайная гауссова матрица с i.i.d. элементами.

Бинарный вектор (скетч) $\mathbf{h}(\mathbf{x})$ также известен как бинаризованное вложение (binary embedding) [14] или SimHash ([15] и подразд. 2.1).

1.1. Искажение оценок угла по бинаризованным гауссовым случайным проекциям. Точность оценки мер расстояния/сходства измеряют искажением (distortion). Мерой искажения рандомизированных вложений в среднем (для несмещенных оценок) является дисперсия оценки. Математическое ожидание (м.о.) E и дисперсию V оценки θ^* угла $\theta(\mathbf{x}, \mathbf{y})$ можно получить по (1), (2), используя м.о. и дисперсию биномиального распределения бернуллиевой переменной, принимающей значение 1 с вероятностью θ / π [16, 17]:

$$\mathrm{E}\{\theta^*\} = \mathrm{E}\{\pi\, \mathrm{dist}_{\mathrm{ham}}\} = \theta, \ \mathrm{V}\{\theta^*\} = \mathrm{V}\{\pi \mathrm{dist}_{\mathrm{ham}}\} = \theta(\pi - \theta) / d, \tag{3}$$

так как $\mathrm{E}\{\mathrm{dist}_{\mathrm{ham}}\} = \theta / \pi$, $\mathrm{V}\{\mathrm{dist}_{\mathrm{ham}}\} = \mathrm{V}\{h(\mathbf{x}) \neq h(\mathbf{y})\} / d = (\theta / \pi)(1 - \theta / \pi) / d$.

Зная евклидовы нормы $||\mathbf{x}||_2$, $||\mathbf{y}||_2$ исходных векторов $\mathbf{x}$, $\mathbf{y}$, можно по оценкам угла получить оценки скалярного произведения и евклидова расстояния $\mathbf{x}$, $\mathbf{y}$:

$$\langle \mathbf{x}, \mathbf{y} \rangle^* = ||\mathbf{x}||_2 ||\mathbf{y}||_2 \cos\theta^*, \ ||\mathbf{x} - \mathbf{y}||_2^{2*} = ||\mathbf{x}||_2^2 + ||\mathbf{y}||_2^2 - 2\langle \mathbf{x}, \mathbf{y} \rangle^*. \tag{4}$$

Дисперсии и м.о. оценок таких «производных» мер вычисляются по (3), (4) методом линеаризации (delta-method) [16, 18–20].

Рассмотрим теперь искажение оценок $\theta(\mathbf{x}, \mathbf{y})$ в худшем случае. Для угла $\theta(\mathbf{x}, \mathbf{y})$ между любыми единичными векторами $\mathbf{x}$, $\mathbf{y}$ (векторами единичной евклидовой нормы $||\mathbf{x}||_2 = 1$, $||\mathbf{y}||_2 = 1$, т.е. векторами на единичной сфере) с вероятностью не менее $1 - \delta = 1 - 2\exp(-2\varepsilon_a^2 d)$ выполняется

$$|\mathrm{dist}_{\mathrm{ham}}(\mathbf{h}(\mathbf{x}), \mathbf{h}(\mathbf{y})) - \theta(\mathbf{x}, \mathbf{y}) / \pi| \leq \varepsilon_a, \tag{5}$$

где $\varepsilon_a \in (0, 1)$ — аддитивное искажение. Для доказательства применяется неравенство Хефдинга (Hoeffding's inequality) для d реализаций бернуллиевой с.в. [21, 14, 22]. Это можно рассматривать как аналог леммы JL для рас-

пределений (DJL) [5–8], однако в классической DJL используют евклидово расстояние как для входных, так и для вложенных векторов, вложение является линейным, а искажение — мультипликативным $1\pm\varepsilon$: $(1-\varepsilon)\,||\,\mathbf{x}-\mathbf{y}\,||_2 \le$ $\le\ ||\,d^{-1/2}\mathbf{R}\mathbf{x}-d^{-1/2}\mathbf{R}\mathbf{y}\,||_2 \le (1+\varepsilon)\,||\,\mathbf{x}-\mathbf{y}\,||_2$ ([5–8] и обзор [1]).

Применяя неравенство Буля (union bound) к $N(N-1)/2$ углам между N исходными единичными векторами, получаем, что все они сохраняются с большой вероятностью при $d=O(\log N/\varepsilon_a^2)$. (Символы O, Ω, Θ используются согласно асимптотической нотации “Big O” [23].) Это аналог классической леммы JL для евклидовых расстояний. Отметим, что преобразования, сохраняющие евклидовы расстояния с $1\pm\varepsilon$-искажением, сохраняют также углы между единичными векторами с $\pm\varepsilon_a$-искажением [24, 14].

Нижняя граница размерности d бинарных скетчей (полученных любым преобразованием, забывчивым по отношению к N векторам), по которым можно оценить угол между всеми парами векторов с искажением не более ε_a с вероятностью $1-\delta$, составляет $d=\Omega\,(\log\,(N/\delta)/\varepsilon_a^2)$ [14]. Для незабывчивых (зависящих от данных) бинарных скетчей нижняя граница $d=\Omega(\log N/\varepsilon_a^2/\log\,(1/\varepsilon_a))$ [14]. (Отметим, что для классической леммы JL плотная нижняя граница $d=\Omega\,(\log N/\varepsilon^2)$ получена в [25].)

Аналоги леммы JL с аддитивным искажением $\pm\varepsilon_a$ существуют для углов между векторами, принадлежащими некоторым непрерывным (бесконечным) множествам. Так, для k-разреженных (с не более чем k ненулевыми компонентами) единичных векторов выполняется (5) при $d=O(k\log(D/\varepsilon_a)/\varepsilon_a^2)$ (аналог свойства ограниченной изометрии RIP, но для бинарных измерений) [21].

Пусть S — множество единичных векторов со средней гауссовой шириной $\omega:\omega(S)=\mathrm{E}\,\{\sup_{x\in S}\langle\,\mathbf{r},\,\mathbf{x}\,\rangle\}$, где $\mathbf{r}\sim\mathrm{Norm}(\mathbf{0},\mathbf{I}_D)$. Для такого S верхняя граница $d=O(\omega^2(S)/\varepsilon_a^6)$ показана в [26] и улучшена до $d=O(\omega^2(S)/\varepsilon_a^4)$ в [27] (см. также [14]). Для подпространств, разреженных векторов и матриц низкого ранга $d=O(\omega^2(S)/\varepsilon_a^2)$ [27] (так, $d=O(D'/\varepsilon_a^2)$ для подпространства размерности D'). Для задачи различения углов по пороговому значению d уменьшается в $1/\varepsilon_a$ раз.

Сравнение этих результатов оценки угла с аддитивным искажением по бинаризованным вложениям после гауссовых i.i.d. случайных проекций с результатами оценки евклидового расстояния с мультипликативным искажением по вещественным вложениям, полученным линейными гауссовыми i.i.d. случайными проекциями (последние представлены в обзоре [1]), для многих типов множеств показывает сходный характер зависимости размерности d от соответствующих искажений оценок (см. также [27]).

1.2. Ускорение формирования бинаризованных вложений. В целях ускорения формирования бинаризованных вложений для оценки угла между исходными векторами вместо плотных гауссовых i.i.d. случайных матриц применяют так называемые структурированные матрицы. Один из типов таких матриц, использующий варианты быстрого вложения JL (например, FJLT [28], другие варианты быстрого вложения JL см. в обзоре [1]), изначально предлагался для быстрых линейных вещественных вложений и обеспечивает выполнение условий лемм JL с близкими к оптимальным параметрами. Эти так называемые матричные конвейеры (pipelines) аппроксимируют результат обычного случайного проецирования последовательным применением матриц, занимающих мало памяти (от $O(D^2)$ до $O(D)$ и менее), и по-

зволяют достичь (общего) времени умножения $O(D \log D)$. Конвейеры с использованием быстрого вложения JL (FJLE) для бинаризованных вложений рассмотрены в подразд. 1.2.1–1.2.3. Другой тип — разреженные матрицы (подразд. 1.2.4).

1.2.1. Комбинация FJLE и гауссовой матрицы. Здесь используют конвейер $\mathbf{\Psi\Phi}$ (с последующей бинаризацией), в котором $\mathbf{\Phi}$ реализует линейное снижение размерности посредством FJLE, а $\mathbf{\Psi}$ — матрица, «подготавливающая» к бинаризации без существенного изменения размерности (i.i.d. гауссова или ее аппроксимация).

В работе [14] исследуется вложение N векторов с использованием $\mathbf{\Phi} = \mathbf{SHD}_R$, где $\mathbf{S}$ — разреженная матрица, случайно выбирающая строки предыдущей, $\mathbf{H}$ — матрица Адамара (см., например, в [1]), $\mathbf{D}_R$ — диагональная матрица Радемахера (со случайными числами из $\{-1,+1\}$ с вероятностью 1/2). Время умножения на $\mathbf{\Phi}$ составляет $O(D \log D)$. В качестве $\mathbf{\Psi}$ используется гауссова i.i.d. матрица или ее аппроксимация набором B матриц с числом строк в каждой d / B: $\mathbf{\Psi} = \mathbf{STD}_R$, где $\mathbf{T}$ — гауссова теплицева матрица [1]. Результат вычисляется как медианное значение B полученных $\mathrm{dist}_{\mathrm{ham}}$. Теплицевы $\mathbf{\Psi}$ при оптимальном $d = O(\log N / \varepsilon_a^2)$ обеспечивают полное время умножения $O(D \log D)$ при $\log N \lesssim \varepsilon_a D^{1/2}$ и несколько худшее при увеличении $\log N$. (Обозначение $X \lesssim Y$ используется, если $X \le CY$ для универсальной константы $C > 0$.) Гауссова $\mathbf{\Psi}$ обеспечивает быстрое умножение при $\log N \lesssim \varepsilon_a^2 D^{1/2}$; время умножения при $d > O(D^{1/2})$ больше, чем у $\mathbf{\Phi}$.

Для вложения непрерывных (бесконечных) множеств с $\omega(S)$ в [27] проанализирован конвейер $\mathbf{\Psi\Phi}$ с бинаризацией, в котором $\mathbf{\Psi}$ есть i.i.d. гауссова матрица, а $\mathbf{\Phi} = \mathbf{SHD}_R$ (либо i.i.d. гауссова матрица). Для искажения ε_a требуется $d = O(\omega^2(S) / \varepsilon_a^6)$ (либо $d = O(\omega^2(S) / \varepsilon_a^4)$). Также представляет интерес исследование варианта конвейера $\mathbf{SHD}_G \mathbf{HD}_R$ из [29] ($\mathbf{D}_G$ — диагональная гауссова i.i.d.).

1.2.2. Непосредственное использование FJLE. В работе [17] проанализирован конвейер sign ($\mathbf{SCD}_R \mathbf{x}$) при $d \le D$, где $\mathbf{C} = \mathrm{circ}\,(\mathbf{r})$ — гауссова циркулянтная матрица (строки формируют циклическим сдвигом гауссова i.i.d. вектора $\mathbf{r}$), при $d > D$ используется несколько различных $\mathbf{C}$. Циркулянтность $\mathbf{C}$ обеспечивает время умножения $O(D \log D)$. Дисперсия оценки угла вследствие циркулянтности $\mathbf{C}$ увеличивается аддитивно на 32α, где $\max\{\| \mathbf{x} \|_\infty / \| \mathbf{x} \|_2, \| \mathbf{y} \|_\infty / \| \mathbf{y} \|_2\} \le \alpha$. Это позволяет доказать аналог леммы JL для угла при $d = O(\log^2 N / \varepsilon_a^2)$ (для малых α).

Также рассматривается обучение матрицы $\mathbf{C}$. Время получения бинаризованных вложений с помощью $\mathbf{C}$ в сотни раз меньше, чем для плотных i.i.d. матриц (см. подразд. 1.1), и до нескольких раз меньше, чем для конвейера [14] (см. подразд. 1.2.1) и билинейных проекций (подразд. 1.3) при сравнимых результатах поиска по сходству (без обучения $\mathbf{C}$). Отметим, что d изменяли от $D / 8$ до D, для различных задач $D = 2000$–50000, $d = D$ значительно улучшало результаты, как и обучение.

Анализ бинаризованного вложения в [30] для N единичных векторов с малым $\alpha = O(D^{-1/2})$ дал $d = O(\log N / \varepsilon_a^3)$ при $\log N \lesssim D^{1/2}$, $d \le D^{1/2}$.

В работе [31] время преобразования уменьшено до $O(D + d \log d)$ за счет отбора сэмплированием d компонентов входного вектора перед умножением на $\mathbf{C}$.

«Короткий» матричный конвейер sign ($\mathbf{MD}_R \mathbf{x}$) в [32] включает «регулярные» (псевдо)случайные матрицы $\mathbf{M}$ с элементами — суммой некоторого числа независимых гауссовых переменных (к ним относятся гауссовы $\mathbf{C}$, $\mathbf{T}$ и ряд других структурированных матриц). Показано увеличение дисперсии оценки угла, концентрация оценок угла для конечных множеств.

1.2.3. Конвейеры со сглаживанием входных векторов. Для работы с векторами без ограничения на α используется предобработка (сглаживание) путем $\mathbf{HD}_R\mathbf{x}$. К этому типу относится «расширенный» матричный конвейер $\text{sign}(\mathbf{MD}_R\mathbf{HD}_R\mathbf{x})$ из [32]. Для конвейера $\text{sign}(\mathbf{SCHD}_R\mathbf{x})$ в [30] показано, что существует возможность ε_a-вложения, не зависящего от величины α, для $d = O(\log N / \varepsilon_a^3)$ при $\log N \lesssim D^{1/3}$, а при $\log N \lesssim D^{1/2}$ только незначительная доля векторов после $\mathbf{HD}_R\mathbf{x}$ не имеет малой нормы $||\mathbf{x}||_\infty \le C(\log D / D)^{1/2}$. Проблема анализа при больших значениях $\log N$, т.е. когда $d = O(D)$, пока не решена.

Необходимость применения сглаживания $\mathbf{HD}_R\mathbf{x}$ на практике неочевидна [30], так как и без него получают хорошие результаты в задачах поиска по сходству и классификации, особенно при использовании обучения $\mathbf{C}$ [17]. В [33] показано, что для $\text{sign}(\mathbf{SCD}_R\mathbf{x})$ при случайном независимом сэмплировании посредством $\mathbf{S}$ требование малости α избыточно.

1.2.4. Ускорение проецирования случайными разреженными матрицами. Ускорение формирования бинаризованных вложений для оценки угла и производных мер (4) с применением проецирования случайными тернарными i.i.d. матрицами, в том числе разреженными (с элементами из $\{-1, 0, +1\}$ с вероятностями $\{q/2, 1-q, q/2\}$), рассмотрено в [34, 18], а с помощью бинарных матриц, в том числе разреженных (с элементами из $\{0, +1\}$ с вероятностями $\{1-q, q\}$), описано в [35, 19]. Экспериментальная дисперсия оценки угла по бинаризованным вложениям при использовании таких матриц близка к теоретической (3), когда распределение компонентов $\mathbf{Rx}$ близко к гауссову; это также справедливо для дисперсии оценки производных мер. Сходимость к гауссову распределению и скорость сходимости с применением неравенства Берри–Эссеена [36] исследовались в [34, 35], но их связь с точностью оценки угла не анализировалась.

Эксперименты [27] по исследованию максимального искажения ε_a с использованием разреженной ($q = 1/3$) гауссовой матрицы показывают практически такие же результаты, как и для плотной гауссовой матрицы.

Таким образом, известные аналитические результаты для оценки угла с аддитивным искажением по быстрым бинаризованным вложениям с применением вариантов FJLE (для множества N векторов) аналогичны результатам, полученным для плотных i.i.d. гауссовых матриц с бинаризацией (см. подразд. 1.1). Отметим также аналогию с результатами по оценке евклидового расстояния (но с мультипликативным искажением) по вещественным FJLE (см. обзор [1]). Темой текущих исследований является получение аналитических результатов с гарантиями худшего случая для быстрых бинаризованных вложений непрерывных (бесконечных) множеств, а также для проецирования разреженными (псевдо)случайными матрицами [33] (обзор Sparse Johnson-Lindenstrauss Transform см. в [1]).

1.3. Формирование бинаризованных вложений большой размерности. Экономия вычислений и памяти для бинаризованных вложений большой размерности d достигается за счет бинарности (подразд. 6.1). Такие векторы востребованы в задачах поиска по сходству и задачах с обучением (разд. 3) для получения более высоких результатов, чем при снижении размерности [37]. Большее значение d позволяет уменьшить искажение оценок и/или увеличить число векторов N в аналогах леммы JL.

Из быстрых преобразований (см. подразд. 1.2) для получения бинарных векторов большой размерности $d = D$ (при D, изменяющемся до 51200) использовались циркулянтные конвейеры ([17]) и подразд. 1.2.2). В режиме без снижения размерности время умножения $O(D \log D)$, требуемая память $O(D)$.

Преобразованием без изменения размерности (и с сохранением евклидовых расстояний) является поворот. Случайное вращение используется для выравнивания дисперсии компонентов вещественного вектора, что полезно для применений в задачах поиска по сходству или линейной классификации как при использовании вещественных векторов, так и при подготовке векторов для бинаризации.

Для быстрой реализации случайных вращений в [37] предложено билинейное случайное проецирование $\mathrm{vec}\,(\mathbf{R}_1\mathbf{X}\mathbf{R}_2)$, где $\mathbf{X}$ — вектор $\mathbf{x}$, преобразованный в матрицу, $\mathbf{R}_1$ и $\mathbf{R}_2$ — случайные ортогональные матрицы. Время умножения $O(D^{3/2})$, требуемая память $O(D)$. Возможно аналогичное преобразование с неортогональными $\mathbf{R}_1$ и $\mathbf{R}_2$. В экспериментах эти преобразования медленнее циркулянтного $\mathbf{C}\mathbf{D}_R\mathbf{x}$ (см. подразд. 1.2) в два-три раза [17].

Ускорить поворот и сэкономить память по сравнению с циркулянтным преобразованием позволяет [38] ортогональная матрица, которая является кронекеровым (тензорным) произведением элементарных ортогональных матриц. Для матриц 2×2 сложность умножения составляет $O(D\log D)$, а хранения — всего лишь $O(\log D)$. Увеличивая размер элементарных матриц, число параметров и время умножения можно варьировать до $O(D^2)$. Аналитическое исследование искажения угла для бинаризованных скетчей в [37, 38] не проводилось. Как и матрицу $\mathbf{C}$ (см. подразд. 1.2), матрицы $\mathbf{R}_1$, $\mathbf{R}_2$ и кронекеровы можно формировать обучением [37, 38].

2. LSH-ФУНКЦИИ И СКЕТЧИ С БИНАРНЫМИ И ДИСКРЕТНЫМИ ЭЛЕМЕНТАМИ

Механизм формирования скетчей с бинарными (или целочисленными) компонентами для оценки мер расстояния/сходства исходных представлений объектов (главным образом, векторных) предоставляют функции локально-чувствительного хэширования (locality sensitive hashing, LSH). В отличие от обычного хэширования, когда хэши неодинаковых объектов стремятся сделать разными, для LSH-функций семейства F вероятность коллизии хэшей объектов x, y [13] зависит от их сходства

$$\Pr_{\mathrm{F}}\{h(x)=h(y)\}=\mathrm{sim}\,(x,\,y), \tag{6}$$

где sim — мера сходства объектов x, y, принимающая значения в $[0,1]$. Примеры LSH приведены в подразд. 2.1. Отметим, что изначально локально-чувствительное хэширование с несколько другой формулировкой было предложено для сублинейного поиска по сходству с помощью хэш-таблиц, адресуемых LSH-хэшами [39–43].

Компоненты d-мерных скетчей получают как значения d LSH-функций $h_i(x)$, $i\in[d]$ (где $[n]$ обозначает множество $\{1,\ldots,n\}$) из параметризованного семейства F со случайно и независимо выбранными параметрами для каждого i. Оценку $\Pr^*$ вероятности $\Pr\{h(x)=h(y)\}$ по d-мерным скетчам получают как

$$\Pr^*\{h(x)=h(y)\}=\sum_{i=1}^{d}1\{h_i(x)=h_i(y)\}/d=1-\mathrm{dist}_{\mathrm{ham}}\,(\mathbf{h}(x),\,\mathbf{h}(y)). \tag{7}$$

Отметим, что sim может также являться монотонно возрастающей функцией f со значениями в $[0,1]$ от некоторой другой функции сходства sim′: $\Pr\{h(x)=h(y)\}=$ $=\mathrm{f}(\mathrm{sim}'\,(x,\,y))$. Аналогично для монотонно убывающей g и расстояния dist′: $\Pr\{h(x)=h(y)\}=\mathrm{g}(\mathrm{dist}'(x,\,y))$. Если протабулировать $\Pr\{h(x)=h(y)\}$ от значений $\mathrm{sim}'\,(x,\,y)$, то по $\Pr^*$ из таблиц можно получить оценки sim′ [18, 44]. Также можно использовать f^{-1}, если она известна. Аналогично оценивается $\mathrm{dist}'(x,\,y)$.

Функции сходства sim, которые задают семейства LSH (6), являются ядерными функциями (разд. 3), т.е. их можно использовать в ядерных методах. Кроме того, оценка $\Pr^*$ (7) в точности равна скалярному произведению бинарных скетчей, полученных позиционным кодированием значений компонентов $\mathbf{h}(x)$ (подразд. 3.3). Поэтому такие бинарные скетчи можно непосредственно применять в быстрых линейных методах (например, в линейном методе опорных векторов SVM [45–49]) с результатами, близкими к результатам ядерных методов с соответствующим исходным (линейным или нелинейным) ядром.

2.1 Примеры LSH-функций. Из определения LSH следует, что бинаризованные вложения для оценки угла (см. разд. 1) являются результатом применения LSH-функций SimHash [13, 15] (для $\mathrm{dist}'(\mathbf{x}, \mathbf{y}) = \theta(\mathbf{x}, \mathbf{y}) / \pi$ с преобразованием $\mathrm{g}: 1 - \mathrm{dist}'(\mathbf{x}, \mathbf{y})$).

Для расстояний Минковского L_s, $s \in (0, 2]$, между вещественными векторами предложены семейства LSH [50] с использованием векторов $\mathbf{r}$ с компонентами — i.i.d. с.в. из s-устойчивых распределений

$$h(\mathbf{x}) = \lfloor (\langle \mathbf{r}, \mathbf{x} \rangle + U) / w \rfloor, \tag{8}$$

где w — параметр, определяющий размер хэш-корзины, U — i.i.d. с.в. из равномерного распределения в $[0, w]$: $U \sim \mathrm{Unif}\,(0, w)$. Примерами s-устойчивых распределений являются распределение Коши для $s = 1$ и Гаусса для $s = 2$.

Для единичных векторов $\mathbf{x}$, $\mathbf{y}$ целочисленные скетчи с компонентами (8), полученные для евклидового расстояния L_2, можно использовать для оценки $\cos(\mathbf{x}, \mathbf{y})$ [44]. Показано, что меньшая дисперсия оценки достигается при $U = 0$, особенно для малых $\cos(\mathbf{x}, \mathbf{y})$. При увеличении w число битов в LSH-коде уменьшается, при $w \geq 3$ практически получают SimHash.

В работах [44, 51] скетчи для оценки $\cos(\mathbf{x}, \mathbf{y})$ конструируют в отличие от равномерного разбиения (8), неравномерно разбивая $\langle \mathbf{r}, \mathbf{x} \rangle$ на интервалы $(-\infty, -w)$, $[-w, 0)$, $[0, w)$, $[w, \infty)$ и кодируя их двумя битами. Оценку $\cos(\mathbf{x}, \mathbf{y})$ по скетчам получают как по эмпирической вероятности совпадения LSH-кодов [44], так и более точно методом максимального правдоподобия MLE по эмпирическим вероятностям в $4 \times 4 = 16$ областях [51].

В работе [52] предложены LSH для расстояния χ^2: $\mathrm{dist}_{\chi^2} = \sum_{i=1}^{D} (x_i - y_i)^2 / (x_i + y_i)$ от положительных векторов, где в (8) $\mathbf{r}$ генерируют из $\mathrm{Norm}^+(0, 1)$ (положительного гауссова).

Бинаризацию s-устойчивых случайных проекций $\langle \mathbf{r}, \mathbf{x} \rangle$ исследуют в [53]. Для распределения Коши ($s = 1$) и неотрицательных векторов ($\mathbf{x}, \mathbf{y} \geq 0$) с единичной нормой L_1 $\Pr\{h(\mathbf{x}) = h(\mathbf{y})\} \approx 1 - (1/\pi) \arccos(\mathrm{sim}_{\chi^2})$, где $\mathrm{sim}_{\chi^2} = (2 - \mathrm{dist}_{\chi^2}) / 2$. В [54] таким же образом полученные бинарные скетчи для разных $0 < s \leq 2$ применяют в задачах обучения (после позиционного кодирования (подразд. 3.3), что дает некоторую нелинейную ядерную функцию исходных $\mathbf{x}$, $\mathbf{y}$).

В работе [55] для ускорения LSH для L_2 (8) используют быстрые матричные конвейеры $\mathbf{GHD}_R$, $\mathbf{HD}_G\mathbf{PHD}_R$, где матрица $\mathbf{G}$ — гауссова разреженная i.i.d., матрица $\mathbf{P}$ — случайной перестановки (см. также подразд. 1.2, [29] и обзор [1]). Показано, что вероятности коллизий близки к исходному LSH.

Недостатком LSH является необходимость разработки LSH-семейств для каждого типа данных и меры расстояния/сходства. В некоторой степени это компенсируется наличием уже разработанного арсенала LSH-семейств (главным об-

разом, для векторов) [41, 56, 43]. Однако каждое новое LSH-семейство можно применять не только для поиска по сходству с использованием хэш-таблиц, но и для формирования скетчей для оценки соответствующего расстояния/сходства.

Рассмотрим подробнее LSH-функции и скетчи на их основе для обобщенного сходства Жаккара — меры сходства (известной также как ядро min-max [57, 58]), которая показывает хорошие результаты в задачах классификации [57, 58].

2.2. Функции LSH для обобщенного сходства Жаккара. Оценка обобщенного сходства Жаккара неотрицательных вещественных векторов ($\mathbf{x}, \mathbf{y} \geq 0$) [13, 59, 60] $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{D} \min(x_i, y_i) / \sum_{i=1}^{D} \max(x_i, y_i)$ (и $||\mathbf{x}-\mathbf{y}||_1$ через соотношение [60] $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y}) = (||\mathbf{x}||_1 + ||\mathbf{y}||_1 - ||\mathbf{x}-\mathbf{y}||_1) / (||\mathbf{x}||_1 + ||\mathbf{y}||_1 + ||\mathbf{x}-\mathbf{y}||_1)$) по целочисленным скетчам на основе LSH для $\mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y})$ исследована в [13, 59–62, 57] и ссылках к ним.

Consistent weighted sampling (CWS) [59] из $\mathbf{x} \geq 0$ генерирует скетч (i_j, t_j), $j = 1, \ldots, d$, $i \in [D]$, с целочисленными компонентами, такой что $\Pr\{h(\mathbf{x}) = h(\mathbf{y})\} = \Pr\{(i_{xj}, t_{xj}) = (i_{yj}, t_{yj})\} = \mathrm{sim}_{\mathrm{JG}}(\mathbf{x}, \mathbf{y})$, т.е. является LSH для $\mathrm{sim}_{\mathrm{JG}}$. Время генерации $O(\mathrm{nnz}(\mathbf{x})d)$, где $\mathrm{nnz}(\mathbf{x})$ — число ненулевых компонентов $\mathbf{x}$, получено в [59] для среднего, а в [60] — для худшего случая. Отметим, что i_j уже содержит информацию о величине компонентов (весах) $\mathbf{x}$. В [57] экспериментально показано, что $\Pr\{(i_{xj}, t_{xj}) = (i_{yj}, t_{yj})\} \approx \Pr\{i_{xj} = i_{yj}\}$. Сокращенный (без t_j) целочисленный хэш (скетч) назван 0-bit CWS [57]. Его бинаризация для применения в линейных методах использует младшие b битов i_{xj} (разд. 4) и их позиционное кодирование (подразд. 3.3). В [58] для таких бинарных скетчей (аппроксимирующих ядро min-max) достигают хороших результатов в задачах классификации при меньшей размерности скетча, чем у вещественных скетчей (аппроксимирующих варианты ядер RBF (разд. 3 и обзор [1])) и чем у бинарных позиционных скетчей для угла (SimHash) и для $1-(1/\pi)\arccos(\mathrm{sim}_{\chi^2})$ (см. подразд. 2.1). Отметим, что ядро min-max не имеет настраиваемых параметров.

Время формирования скетча в [62] улучшено до $O(d)$ в среднем, а каждое хэш-значение занимает всего лишь пять–девять битов. Оценка $\mathrm{sim}_{\mathrm{JG}}$ не по бинарным, а по вещественным скетчам, полученным сэмплированием, рассмотрена в [63] и в обзоре [1].

В работе [64] дано обобщение $\mathrm{sim}_{\mathrm{JG}}$ (названное generalized min-max, GMM) на векторы, компоненты которых могут принимать как положительные, так и отрицательные значения, посредством замены компонента $x > 0$ на пару компонентов $[x; 0]$, а компонента $x \leq 0$ — на $[0; -x]$. К таким векторам размерности $2D$ применяют 0-bit CWS для аппроксимации GMM. В [65] приведен теоретический анализ $\mathrm{sim}_{\mathrm{GMM}}$ и сравнение с $\mathrm{sim}_{\cos}$, а в [66] — аппроксимация $\mathrm{sim}_{\mathrm{GMM}}$ с помощью вещественных векторов, полученных методом Нистрема.

Связь LSH с ядерными функциями рассмотрена в подразд. 3.3, а некоторые LSH для исходных бинарных векторов — в разд. 4.

3. АППРОКСИМАЦИЯ ЯДЕРНЫХ СХОДСТВ ПО БИНАРНЫМ ВЕКТОРАМ

Специальным видом функции сходства является ядерная функция (ядро) $\kappa(x, y)$ [67–69]. 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 (обычно полиномиальная) зависит от конкретного типа ядра. Например, для векторов $\mathbf{x}, \mathbf{y}$: $\kappa(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$ — линейное ядро, $\kappa(\mathbf{x}, \mathbf{y}) = \exp(-1/2 \,||\, \mathbf{x} - \mathbf{y} \,||\, {}^2_2 / \sigma^2)$ — гауссово RBF. Другим примером являются ядра для структурированных объектов (графов и др.), основанные на сходстве их подструктур как с явным представлением последних $\boldsymbol{\varphi}$ (например, [70–73]), так и без него, с непосредственным вычислением значения ядра по графам (например, [74–77]).

Быстрое вычисление или оценка значений ядерных сходств полезны для многих приложений, например, для поиска по сходству, а также для ядерных методов (таких, как метод опорных векторов SVM [67–69]), оперирующих только с $\kappa(x, y)$ без использования нелинейного $\boldsymbol{\varphi}(x)$ (так называемый «kernel trick»). Недостаток обучения в ядерных методах — необходимость вычисления (и использования) N^2 ядерных сходств между N объектами базы (т.е. матрицы ядра $\mathbf{K}$), для больших N хранение $\mathbf{K}$ невозможно. Результатом обучения является вычисление функции вида $\mathrm{f}(x) = \sum_{i=1}^{n} a_i \kappa(x, x_i)$, где x_i — некоторые опорные объекты (ОО). Вычисление $\mathrm{f}(x)$ затратное, так как число n ОО может быть велико.

Подходы к преодолению недостатков ядерных методов рассмотрены в обзоре [1]. Они включают непосредственное (без перехода к $\boldsymbol{\varphi}(x)$, $\boldsymbol{\varphi}(y)$) формирование векторов $\boldsymbol{\psi}(x)$, $\boldsymbol{\psi}(y)$ умеренной размерности d таких, что $\kappa(x, y) \approx$ $\approx \langle \boldsymbol{\psi}(x), \boldsymbol{\psi}(y) \rangle / d$. Алгоритмы, работающие с этими векторами, могут быть более эффективными, чем ядерные. Например, для линейной модели $\mathrm{f}(x) = \langle \boldsymbol{\psi}(x), \mathbf{w} \rangle$, $\mathbf{w} = \sum_{i=1}^{n} a_i \boldsymbol{\psi}(x)$, т.е. $\mathrm{f}(x)$ вычисляется за время $O(d)$, затраты памяти на хранение модели тоже $O(d)$. В то же время результаты ядерного SVM с ядром κ близки к результатам линейного SVM [45–49], оперирующего с аппроксимирующими κ векторами $\boldsymbol{\psi}$.

После публикации [78] получили распространение подходы к забывчивому формированию векторных представлений (вещественных и бинарных) для аппроксимации ядер, которые можно представить в виде

$$\kappa(x, y) = \mathrm{E}_{\mathbf{r}} \{\psi(x, \mathbf{r}) \psi(y, \mathbf{r})\}, \tag{9}$$

где $\mathbf{r}$ — случайный вектор параметров из некоторого распределения, зависящего от κ. Для точного вычисления $\kappa(x, y)$ требуются $\boldsymbol{\psi}(x)$, $\boldsymbol{\psi}(y)$ бесконечной размерности, а при оценке точность растет с увеличением размерности d векторов $\boldsymbol{\psi}(x)$, $\boldsymbol{\psi}(y)$. Подходы к формированию вещественных $\psi(\mathbf{x}, \mathbf{r})$ для некоторых типов ядер от векторных представлений $\kappa(\mathbf{x}, \mathbf{y})$ (например, инвариантных к сдвигу ядер типа RBF и др.) называют random feature map, RFM ([78] и обзор [1]).

В подразд. 3.1 рассмотрена бинаризация RFM для инвариантных к сдвигу ядер, в подразд. 3.2 — формирование бинарных скетчей для аппроксимации углового сходства между векторами вторичного пространства, индуцированного любыми нормированными ядрами, в подразд. 3.3 — использование LSH-функций для формирования бинарных скетчей, аппроксимирующих ядерные сходства.

3.1. Аппроксимация инвариантных к сдвигу ядер по бинарным векторам. Для инвариантных к сдвигу ядер $\kappa(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{x} - \mathbf{y})$ (гауссова RBF, лапласова, Коши и др.) согласно теореме Бохнера [78] имеется векторное представление (9).

Например, для RBF $\psi(\mathbf{x}, \mathbf{r}, U) = \sqrt{2}\cos(\langle \mathbf{x}, \mathbf{r}\rangle + U)$, где $\mathbf{r} \sim \mathrm{Norm}(\mathbf{0}, \mathbf{I}/\sigma^2)$, $U \sim \mathrm{Unif}(0, 2\pi)$ ([78] и обзор [1]).

В работе [79] случайные признаки RFM $\psi(\mathbf{x}, \mathbf{r}, U) = \sqrt{2}\cos(\langle \mathbf{x}, \mathbf{r}\rangle + U)$ бинаризуют с помощью квантователя $Q_\xi(\psi(\mathbf{x})) = \mathrm{sign}(\psi(\mathbf{x}) + \xi)$, $\xi \sim \mathrm{Unif}(-1, +1)$. Тогда по $\mathrm{dist}_{\mathrm{ham}}$ между полученными d-мерными бинарными скетчами можно с аддитивным искажением оценить функцию от $1 - \kappa(\mathbf{x} - \mathbf{y})$ как для множества N векторов, так и для компактного подмножества векторов.

Близкие теоретические и экспериментальные результаты получены в [80] для бинаризованных гауссовых билинейных проекций ($\mathbf{R}_1\mathbf{X}\mathbf{R}_2$ вместо $\langle \mathbf{x}, \mathbf{r}\rangle$ (см. подразд. 1.3)).

3.2. Ядерное локально-чувствительное хэширование KLSH. В [81] формируют бинарные LSH-хэши $h(x) = \mathrm{sign}(\langle \boldsymbol{\psi}(x), \mathbf{r}\rangle)$, для которых $\Pr\{h(x) = h(y)\} \approx 1 - (1/\pi)\arccos\kappa(x, y)$. Здесь κ — произвольное ядро, нормированное к [0, 1], вектор $\boldsymbol{\psi}(x)$ — результат проекции вектора $\boldsymbol{\varphi}(x)$ из вторичного пространства H, индуцированного ядром κ, в подпространство, соответствующее главным компонентам PCA, $\mathbf{r}$ — гауссов вектор в этом подпространстве [82].

Реально вычисление $h(x)$ проводится с использованием центрированной матрицы ядра $\mathbf{K}(n \times n)$ и $\kappa(x, x_j)$ как $h(x) = \mathrm{sign}\left(\sum_{j=1}^{n} w_j \kappa(x, x_j)\right)$, где для каждой из d функций $h(x)$ вычисляется вектор $\mathbf{w} = \mathbf{K}^{-1/2}\mathbf{e}$, $\mathbf{e}$ — вектор со случайно выбранными для каждой из $h(x)$ единичными компонентами из $[n]$ (см. [81, 82]).

Анализ средней точности аппроксимации ядра для KLSH методом Нистрема приведен в [83], а критические замечания к нему, отличия от метода Нистрема и анализ для поиска по сходству — в [82].

3.3. Формирование бинарных векторов с использованием LSH для оценки ядерных сходств. Как упоминалось в разд. 2, семейство LSH задает функцию сходства sim (6), которая является ядерной. Это следует из возможности представления sim в виде (9) (что эквивалентно скалярному произведению векторов бесконечной размерности с соответствующей нормировкой) бинарным позиционным кодированием $h(x)$.

Пусть $h(x)$ может принимать 2^B значений. Тогда для представления $h(x)$ используют 2^B битов с ровно одним единичным битом в соответствующей значению позиции ([42, гл. 4], подход хэш-корзин в [78], [84]). Например, при $B = 1$ значение 1 кодируется как «10», а 0 (или –1) — как «01». Тогда вычислить число совпадающих компонентов d-мерных скетчей $\mathbf{h}(x)$ (для оценки вероятности Pr (6), т.е. для аппроксимации значения ядра κ (9)) можно как скалярное произведение соответствующих им бинарных векторов фиксированной размерности $d2^B$ с ровно d единичными компонентами, полученными позиционным кодированием. Ошибка аппроксимации κ уменьшается при увеличении d. Кроме того, такие бинарные векторы можно непосредственно использовать в линейных моделях (например, в линейном SVM [45–49]), с результатами обучения, близкими к результатам ядерных методов с соответствующим ядром, но с меньшими затратами памяти и вычислений (для больших N и умеренных d). Для уменьшения числа битов 2^B на представление значения $h(x)$ в [42, гл. 4] применяют хэширование $[2^B] \to \{-1, +1\}$, что несколько уменьшает точность оценок при одинаковом d. В [84] используют позиционное кодирование только младших b битов $h(x)$ для задач линейного обучения (см. также b-битовое кодирование в подразд. 4.2).

Примерами LSH-семейств, точно соответствующих (т.е. $\Pr_{\mathrm{F}}\{h(x)=h(y)\}=\kappa(x, y)$) ядерной функции $\kappa(x, y)$ простого вида, являются: SimHash для $\kappa(\mathbf{x}, \mathbf{y})=1-\theta(\mathbf{x}, \mathbf{y})/\pi$ (см. разд. 1 и подразд. 2.1), MinHash для $\kappa((\mathbf{x}, \mathbf{y})=\mathrm{sim}_{\mathrm{J}}(\mathbf{x}, \mathbf{y})$ (разд. 4). Для расширения типов аппроксимируемых ядер в [42, гл. 4] предложены семейства ε-приближенных хэш-функций ядра (K-kernel hash functions, KHF), для которых $|\Pr\{h(x)=h(y)\}-\kappa(x, y)|\le\varepsilon_a$. Используя сокращение памяти на KHF-скетчи хэшированием $h_i\to\psi_i\colon[2^B]\to\{-1, +1\}$, $i=1, \ldots, d$, при $d=O(\log(1/\delta)/\varepsilon_a^2)$ получают $|\langle\boldsymbol{\psi}(x), \boldsymbol{\psi}(y)\rangle/d-\kappa(x, y)|\le 2\varepsilon_a$ с вероятностью не менее $1-\delta$. Примеры ε-приближенных KHF (на основе известных LSH-семейств) для ядра лапласиана $\exp(-||\mathbf{x}-\mathbf{y}||_1/\sigma)$, а также для ядра $\exp(-||\mathbf{x}-\mathbf{y}||_2/\sigma)$, близкого к RBF, и других ядер описаны в [42, гл. 4].

4. БИНАРНЫЕ И ЦЕЛОЧИСЛЕННЫЕ СКЕТЧИ ДЛЯ ОЦЕНКИ СХОДСТВА БИНАРНЫХ ВЕКТОРОВ

Скетчи и вложения (как вещественные, так и бинарные) для оценки мер расстояния/сходства вещественных векторов можно применять для оценки этих же мер бинарных векторов. Так, скетчи, полученные сэмплированием, позволяют оценить любые линейные суммирующие статистики (см. обзор [1]). Скетчи бинарных векторов с компонентами из {0, 1} на основе LSH и сэмплирования для оценки расстояния Хэмминга представлены в подразд. 4.1, а сходства Жаккара — в подразд. 4.2.

Отметим, что бинарные векторы можно рассматривать как характеристические векторы соответствующих невзвешенных множеств и вычислять меры сходства/различия таких множеств. Так, сходство Жаккара бинарных векторов $\mathbf{x}$, $\mathbf{y}$ определяется как $\mathrm{sim}_{\mathrm{J}}(\mathbf{x}, \mathbf{y})=\langle\mathbf{x}, \mathbf{y}\rangle/(|\mathbf{x}|+|\mathbf{y}|-\langle\mathbf{x}, \mathbf{y}\rangle)$, где $|\mathbf{x}|$ — число ненулевых компонентов $\mathbf{x}$, $\langle\mathbf{x}, \mathbf{y}\rangle\equiv|\mathbf{x}\ \mathrm{AND}\ \mathbf{y}|$, AND — побитовая операция «И». Для множеств X, Y с характеристическими векторами $\mathbf{x}$, $\mathbf{y}$ (1 соответствует наличию элемента множества, 0 — отсутствию): $\mathrm{sim}_{\mathrm{J}}(\mathrm{X}, \mathrm{Y})=|\mathrm{X}\cap\mathrm{Y}|/|\mathrm{X}\cup\mathrm{Y}|=\mathrm{sim}_{\mathrm{J}}(\mathbf{x}, \mathbf{y})$, где $\cap$ — пересечение, $\cup$ — объединение множеств, $|\mathrm{X}|$ — мощность множества X. Кроме того, многие меры расстояния/сходства бинарных векторов и множеств можно вычислить (или аппроксимировать) по другим, например, $\mathrm{dist}_{\mathrm{Ham}}(\mathbf{x}, \mathbf{y})=|\mathbf{x}|+|\mathbf{y}|-2\langle\mathbf{x}, \mathbf{y}\rangle=||\mathbf{x}-\mathbf{y}||_s^s=|\mathrm{X}\cup\mathrm{Y}|-|\mathrm{X}\cap\mathrm{Y}|$ и т.п.

4.1. Скетчи для оценки расстояния Хэмминга. Рассмотренная в [6, 40] LSH-функция для расстояния Хэмминга случайно выбирает один из компонентов вектора $\mathbf{x}$: $h(\mathbf{x})=x_i$, реализуя таким образом простое случайное сэмплирование с замещением (см. обзор [1]). Для этой LSH-функции $\Pr\{h(\mathbf{x})=h(\mathbf{y})\}=\Pr\{x_i=y_i)\}=1-\mathrm{dist}_{\mathrm{Ham}}(\mathbf{x}, \mathbf{y})/D$. Оценка $\mathrm{dist}^*_{\mathrm{Ham}}(\mathbf{x}, \mathbf{y})=(D/d)\mathrm{dist}_{\mathrm{Ham}}(\mathbf{h}(\mathbf{x}), \mathbf{h}(\mathbf{y}))$, ее дисперсия при $d<<D$ [85]:

$$\mathrm{V}\{\mathrm{dist}^*_{\mathrm{Ham}}(\mathbf{x}, \mathbf{y})\}=(D/d)^2 d\,\mathrm{V}\{\mathrm{dist}_{\mathrm{Ham}}(h(\mathbf{x}), h(\mathbf{y}))\}=$$

$$=(D^2/d)(\mathrm{dist}_{\mathrm{Ham}}(\mathbf{x}, \mathbf{y})/D-(\mathrm{dist}_{\mathrm{Ham}}(\mathbf{x}, \mathbf{y})/D)^2).$$

Для $h(\mathbf{x})$, которая является конкатенацией m случайно выбранных компонентов $\mathbf{x}$, $\Pr\{h(\mathbf{x})=h(\mathbf{y})\}=(1-\mathrm{dist}_{\mathrm{Ham}}(\mathbf{x}, \mathbf{y})/D)^m$.

Формировать скетчи для $\mathrm{dist}_{\mathrm{Ham}}$ посредством умножения исходного бинарного вектора на случайную i.i.d. бинарную матрицу (с элементами 1 с вероятностью q и 0 с вероятностью $1-q$) и бинаризации компонентов скетча делением по модулю 2 предложено в [86] (см. также [85]). При этом $\Pr\{h(\mathbf{x})\ne h(\mathbf{y})\}=$

$= 0.5(1-(1-2q)^{\mathrm{dist}_{\mathrm{Ham}}(\mathbf{x},\mathbf{y})})$. Недостатком является необходимость подбора q, минимизирующего дисперсию оценки, для конкретных $\mathrm{dist}_{\mathrm{Ham}} / D$.

Отметим, что для разреженных бинарных векторов применение сэмплирования только ненулевых компонентов и без замещения позволяет уменьшить дисперсию оценки линейных суммирующих статистик, к которым относится расстояние Хэмминга [87, 88].

4.2. Скетчи для оценки сходства Жаккара на основе LSH MinHash. Идея min-скетчей [89, 90] (см. также [91]) для невзвешенных множеств (или их бинарных характеристических векторов размерности D) заключается в присвоении каждому элементу множества (компоненту вектора) некоторого ранга или хэш-значения и в отборе в скетч (запомнив идентификатор ID) элемента множества (ненулевого компонента вектора) с минимальным рангом.

В работах [92–94] такой (целочисленный) скетч MinHash применен для оценки $\mathrm{sim}_{\mathrm{J}}$. Для идеальной (perfect) хэш-функции h, которая не дает коллизий и обеспечивает одинаковую вероятность $1/|S|$ минимального ранга каждого элемента множества S, $\Pr\{\min h(\mathbf{x}) = \min h(\mathbf{y})\} = \mathrm{sim}_{\mathrm{J}}(\mathbf{x}, \mathbf{y})$ (такой хэш-функцией является случайная перестановка). Поэтому MinHash есть LSH.

Для уменьшения дисперсии используют скетчи с d компонентами:

$$\mathrm{V}\{\mathrm{sim}^{*}_{\mathrm{J}}(\mathbf{x}, \mathbf{y})\} = (1/d)\mathrm{sim}_{\mathrm{J}}(\mathbf{x}, \mathbf{y})(1-\mathrm{sim}_{\mathrm{J}}(\mathbf{x}, \mathbf{y})).$$

Так как D может достигать 2^{64} [94], представление ID компонента скетча требует $B = 64$ бита. Для экономии памяти в [85] предложено значительно сократить B (до $b = 1$ или 2) отбрасыванием старших битов. Дисперсия оценки $\mathrm{sim}_{\mathrm{J}}$ по таким b-битовым скетчам [95] $\mathrm{V}\{\mathrm{sim}^{b*}_{\mathrm{J}}(\mathbf{x}, \mathbf{y})\} = (1-\mathrm{sim}_{\mathrm{J}}(\mathbf{x}, \mathbf{y})(\mathrm{sim}_{\mathrm{J}}(\mathbf{x}, \mathbf{y}) + 1/(2^{b}-1))/d$. При $\mathrm{sim}_{\mathrm{J}} > 0.5$ сравнимую с 64-битовыми скетчами точность оценки по однобитовым скетчам удается получить при экономии (битов) памяти в 20 и более раз [85]. Использование MLE-оценок позволяет до 100 раз уменьшить дисперсию для малых $\mathrm{sim}_{\mathrm{J}}$ и больших $\langle \mathbf{x}, \mathbf{y} \rangle / |\mathbf{x}|$. Для применения в задачах обучения линейных моделей [84] скетчи бинаризуют позиционным кодированием (см. подразд. 3.3).

В работе [95] компактные бинарные скетчи для оценки $\mathrm{sim}_{\mathrm{J}}$ формируют из хэшей MinHash, используя еще одну хэш-функцию, которая отображает значения MinHash в [d], причем значение бита при коллизии изменяется (с нуля на единицу, с единицы на нуль). Оценка $\mathrm{sim}_{\mathrm{J}}$ проводится нелинейной функцией числа единиц в векторе XOR скетчей. Точность таких скетчей превосходит точность b-битовых MihHash скетчей при тех же затратах памяти и стоимости хэширования для $\mathrm{sim}_{\mathrm{J}} > 0.75$. Отметим, что эти скетчи пригодны для оценки любых сходств, соответствующих LSH-функциям (6).

При d порядка сотен и тысяч невозможно хранить d перестановок для больших D. Эмуляция перестановки хэш-функцией, в простейшем виде $h(x) = ax + b \bmod \mathrm{prime}$ (prime — простое число), дает малое число коллизий (например, при $h(x) \in [\mathrm{D}^3]$). Однако использование $\Theta(\log 1/\varepsilon)$-независимой хэш-функции для формирования скетчей приводит к неустранимому с ростом d смещению (bias) (до $\pm\varepsilon\,\mathrm{sim}_{\mathrm{J}}$) оценок $\mathrm{sim}_{\mathrm{J}}$ в худшем случае [94, 96–99] (т.е. для данных с низкой энтропией [100]). Сложность вычисления k-независимой хэш-функции также велика: $\Theta(k)$. Малое ε обеспечивают быстро вычислимые хэш-функции на основе табуляции (twisted tabulation) [98, 99], однако необходимость эмуляции d перестановок все равно замедляет получение min-скетча.

Для сокращения числа перестановок можно получать скетчи одной перестановкой (или ее эмуляцией). Такой bottom-скетч формируют из d ненулевых компонентов вектора с минимальными значениями ранга (хэша) [90, 92, 87, 88]. При больших d использование для эмуляции перестановки всего лишь 2-независимых хэш-функций позволяет достичь малого смещения (и дисперсии) [63].

Однако в bottom-скетчах совпадающие ID компонентов, использующиеся для вычисления $\mathrm{sim}_{\mathrm{J}}^{*}$, могут находиться в несоответствующих компонентах скетчей. Поэтому bottom-скетчи нельзя в компактном виде использовать для обучения линейных моделей и в качестве LSH в отличие от b-битовых min-скетчей и partition-скетчей (см. также [88] и обзор [1]).

В работе [101] векторы после одной перестановки равномерно разбивают на d частей (корзин) и ненулевые компоненты с наименьшими номерами в каждой корзине образуют скетч. Такие partition-скетчи предложены для других задач в [89] (см. также [102, 103, 91]). Для LSH-хэширования применяют b-битовое представление, а для линейного обучения — его же в сочетании с позиционным кодированием [101, 104, 105]. Проблемы вследствие возможного отсутствия ненулевых компонентов в некоторых из d частей преодолевают использованием значения ближайшего соседнего ненулевого компонента скетча [105]. Для partition-скетчей в [106] показано, что хэш-функции, основанные на табуляции (mixed tabulation), дают концентрационные границы оценки $\mathrm{sim}_{\mathrm{J}}$, близкие к полностью случайному хэшированию.

В работе [107] формируют бинарные скетчи для оценки $\mathrm{sim}_{\mathrm{J}}$ посредством случайного проецирования разреженными матрицами.

5. ДРУГИЕ НАПРАВЛЕНИЯ ИССЛЕДОВАНИЙ

5.1. Разреженные бинарные векторы для оценки сходства. Ряд алгоритмов представления и обработки данных требуют применения разреженных бинарных векторов (с компонентами из $\{0, 1\}$ и малой долей единичных компонентов), сохраняющих сходство представляемых объектов. Такие разреженные бинарные векторы используются, например, для формирования распределенного представления данных различного типа в ассоциативно-проективных нейронных сетях [108–110]. С разреженными бинарными векторами эффективно работает распределенная автоассоциативная память (Willshaw и Hopfield) [111–113]. Эффективный аналог линейного SVM для разреженных бинарных векторов предложен в [114].

Кроме того, некоторые вычислительные средства специализированы для обработки разреженных бинарных векторов большой размерности, например, ассоциативно-проективные нейрокомпьютеры [108, 115, 116] или инфраструктура поисковых систем [117]. Разреженные распределенные представления также считают нейробиологически обоснованными [118, 119].

В работах [120, 121] для оценки сходства текстовых данных предложено формировать компоненты векторов с регулируемой разреженностью посредством порогового преобразования с ненулевыми порогами (вместо нулевого порога sign (1), (2) в SimHash). Для оценки косинуса угла исходных вещественных векторов в [18, 19, 35] использованы эмпирические вероятности совпадения единичных компонентов бинарных векторов с регулируемой разреженностью, формируемых как $h_{\mathbf{r}}(\mathbf{x}) = 1$ при $\langle \mathbf{r}, \mathbf{x} \rangle \geq t$, $h_{\mathbf{r}}(\mathbf{x}) = 0$ при $\langle \mathbf{r}, \mathbf{x} \rangle < t$, $t \geq 0$. Для случайного проецирования применяют тернарные [18] или бинарные [19, 35] i.i.d. случайные матрицы $\mathbf{R}$ (как неразреженные, так и разреженные). Скалярное произведение полученных бинарных векторов аппроксимирует монотонно убыва-

ющую функцию угла исходных векторов [18], т.е. соответствующее ядро. Аналогично можно преобразовывать векторы, точно представляющие или аппроксимирующие ядра скалярным произведением (см. разд. 3 и, например, [122, 70–73]), для создания (или аппроксимации) новых ядер.

Применение бинарных разреженных векторов, полученных пороговым преобразованием результатов случайного проецирования гауссовой i.i.d. случайной матрицы, для ускорения поиска по сходству (подразд. 6.2) по косинусу угла входных векторов показано в [117], там же приведены близкие результаты экспериментов с использованием структурированных матриц.

В работах [123, 124] бинарные разреженные векторы, по которым можно оценить сходство представляемых ими числовых векторов, формируют «композиционными» методами из бинарных разреженных векторов, отражающих сходство соответствующих компонентов исходных вещественных векторов, с использованием процедуры связывания контекстно-зависимым прореживанием [125]. Показана, в частности, возможность оценки манхэттенова расстояния по бинарным скетчам для некоторых предложенных процедур.

Бинарные разреженные векторы, формируемые с помощью гиперпрямоугольных рецептивных полей RSC [126, 127], позволяют оценивать соответствующее ядерное сходство и решать задачи нелинейной классификации. Аппроксимация ядра RBF для векторов единичной нормы скалярным произведением бинарных разреженных векторов предложена в [128].

В работах [125, 130–132] рассмотрены бинарные разреженные векторные представления графов — эпизодов баз знаний, применяемых при моделировании рассуждений по аналогии ([129] и ссылки к ней). Показано, что сходство эпизодов, определяемое по векторам, соответствует сходству, воспринимаемому человеком [130] при поиске аналогичных эпизодов. Определяемое по векторам сходство можно также использовать для нахождения соответствующих фрагментов двух эпизодов (analogical mapping) [131, 132].

5.2. Вложения с небинарным квантованием. Вложения вещественных векторов в векторы с дискретными компонентами, формируемыми как $\mathbf{Q}(\mathbf{Rx}+\xi)$, исследованы в [22, 133–136]. Здесь $\mathbf{R}$ — i.i.d. гауссовы [22, 135, 136], субгауссовы [133] или другие RIP-матрицы [134], ξ — вектор «размывания» (с i.i.d. равномерными компонентами), $\mathbf{Q}(\cdot)$ — покомпонентный квантователь: обычный многобитовый равномерный скалярный (где B битов кодируют 2^B градаций, [22, 133–135]) или немонотонный универсальный (universal quantization) ([135, 136] и ссылки к ним), который можно рассматривать как сохраняющий только b младших битов равномерного квантователя.

В отличие от вещественных вложений с мультипликативным искажением согласно леммам (D)JL (см. подразд. 1.1) и бинаризованных вложений с аддитивным искажением согласно аналогам лемм (D)JL (см. подразд. 1.1) искажение вложений с небинарным квантованием включает обе составляющие. Аддитивное искажение стремится к нулю при увеличении точности (шага) квантования, а при увеличении размерности вложения d уменьшается как мультипликативное, так и аддитивное искажение.

В работе [22] рассмотрено вложение N вещественных векторов с евклидовым расстоянием L_2 в векторы с дискретными компонентами и манхэттеновым расстоянием L_1. Для этих же функций расстояния в [133] исследуют субгауссовы i.i.d. случайные проекции непрерывных множеств. В [134] аналогичные результаты получены для RIP-матриц $\mathbf{R}$ с возможностью быстрой реализации умножения (подразд. 1.2 и обзор в [1]), для всех множеств векторов, для которых выполняется RIP.

Для универсального квантования [135, 136] показано, что расстояние между вложениями аппроксимирует кусочно-линейную функцию исходного расстояния g (dist). При этом малые расстояния аппроксимируются более точно, что актуально для поиска по сходству (подразд. 6.2). В работе [136] проанализирована также аппроксимация (с аддитивным искажением) ядер, зависящих от расстояний, скалярным произведением векторов дискретных вложений.

6. ОБСУЖДЕНИЕ

Применение рассмотренных в настоящем обзоре векторных представлений с бинарными (или в меньшей степени целочисленными) компонентами в ряде случаев позволяет повысить экономичность хранения и эффективность обработки данных по сравнению с вещественными векторами [1] как при быстрой оценке мер расстояния/сходства исходных объектов, так и при непосредственном использовании векторов для поиска по сходству, обучения и в других задачах.

6.1. Эффективность бинарных и целочисленных векторов. Представление и хранение одного компонента бинарного вектора требует одного бита вместо 32–64 бит для вещественного. При этом точность оценок сходства в терминах числа компонентов для бинарных векторов может быть незначительно ниже или превосходить точность оценок для вещественных векторов. Например, точность оценки скалярного произведения по SimHash всего лишь в 2.5 раза ниже по сравнению с вещественными случайными проекциями при малых значениях $\mathrm{sim}_{\cos}$ [16], а по 1-битовому MinHash — даже выше [84]. Поэтому при одинаковых затратах памяти оценки по бинарным (и целочисленным) векторам могут быть точнее (при некоторых параметрах до 10 [16] и до 10–10000 раз [84]). Для b-битового MinHash при $b=1$ точность оценки $\mathrm{sim}_J\ (>0.5)$ при одинаковых затратах памяти более чем в 20 раз выше, чем при $b=64$ [85] (см. подразд. 4.2). В задачах линейного обучения (с применением эффективного линейного SVM) при использовании позиционного кодирования бинарных и целочисленных скетчей [84, 85] также наблюдается большая экономия памяти и времени обучения при сходных результатах классификации.

Оценка расстояний/сходств исходных объектов по соответствующим бинарным векторам часто требует вычисления расстояния Хэмминга. При представлении одного компонента вектора одним битом (например, 64-разрядного слова) вычисление выполняется посредством быстрых побитовых операций XOR (одновременно над 64 компонентами двух векторов) и подсчетом числа единичных битов в каждом полученном слове. Для подсчета используют быстрые специализированные команды процессора или таблицы. Ускорение вычисления мер расстояния/сходства бинарных векторов по сравнению с вещественными векторами той же размерности может достигать десятков раз. Операции векторно-матричного умножения с использованием бинарного вектора/матрицы также выполняются более эффективно, так как сводятся к сложению (вместо умножения с плавающей точкой). Кроме того, для бинарных векторов возможна эффективная аппаратная поддержка обработки. Некоторые алгоритмы и средства, специализированные для работы с разреженными бинарными векторами, приведены в подразд. 5.3.

6.2. Бинарные векторы в приближенном поиске по сходству. Поиск по сходству — это поиск сходных с объектом-запросом объектов базы по некоторой мере расстояния/сходства. Линейный поиск по сходству (т.е. поиск с использованием оценки сходства между объектом-запросом и всеми объектами базы) по бинарным векторам позволяет за счет эффективности оперирования бинарными векторами уменьшить время линейного поиска по исходным представлениям и мере сходства, хотя и является приближенным вследствие неточности оценок.

Кроме того, быстро полученные результаты можно затем уточнить вычислением точного сходства по исходным представлениям.

Для поиска по сходству обычно требуется оценивать соотношение расстояний/сходств объектов (больше или меньше), причем не во всем их диапазоне, а для значений, соответствующих ближайшим/ближним соседям. Это можно использовать для конструирования более компактных скетчей, обеспечивающих сохранение качества поиска по сходству.

Скетчи, позволяющие увеличивать точность оценки сходства для больших его значений, уже упоминались и в настоящем обзоре [86, 85, 95, 135, 136]. В работах [137–139] предложены методы формирования бинарных скетчей, дающие более точную оценку манхэттенова расстояния, евклидова расстояния, косинуса угла между близкими вещественными векторами по расстоянию Хэмминга между скетчами для применения в приближенном поиске по сходству. В [139] для оценки этих расстояний предложено асимметричное расстояние между двумя скетчами, где в дополнение к бинарным представлениям двух скетчей используется также исходное вещественное представление одного из соответствующих им векторов (при поиске по сходству — вектора запроса). Это позволило на 10–40 % уменьшить размер скетчей при той же точности поиска по сходству.

Гарантии поиска по сходству для исходного расстояния — угол между вещественными векторами — по $\mathrm{dist}_{\mathrm{Ham}}$ бинарных скетчей SimHash ([13] и см. разд. 1) анализируют в [140] (при условии, что база имеет известное число объектов, угол между которыми менее заданного). В [117] исследуют гарантии поиска по сходству по разреженным бинарным скетчам для косинуса угла между исходными вещественными векторами как меры сходства (см. подразд. 5.1).

Быстрая оценка расстояний также ускоряет приближенный поиск по сходству с использованием имеющихся алгоритмов (индексных структур), работающих на основе вычисления расстояний [141–143].

Векторы с бинарными или целочисленными компонентами, формируемыми LSH-функциями, разработанными для конкретных мер расстояния/сходства исходных векторов, непосредственно используют для приближенного сублинейного поиска по соответствующим этим LSH-функциям мерам расстояния/сходства в табличных структурах [39–43]. Векторы с вещественными компонентами нельзя непосредственно применять для поиска с помощью хэш-таблиц. Отметим также подход locality-sensitive filtering [144] (развитие LSH), где для сублинейного поиска по сходству используют фильтрацию векторов по условию $\langle \mathbf{r}, \mathbf{x} \rangle \geq t$ (см. подразд. 5.1).

Для бинарных векторов (размерности от десятков до сотен) разработаны индексные структуры для эффективного поиска по расстоянию Хэмминга с сублинейной сложностью (например, [13, 15, 145–150] и ссылки к ним) как на основе таблиц [13, 15, 146–149], так и деревьев [145] и графов соседства [150]. Некоторые структуры предназначены для приближенного поиска (например, [13, 145, 150]), другие позволяют точный поиск [15, 146–149]. Предложены структуры для поиска как с фиксированным радиусом запроса [15, 146, 149], так и с изменяющимся [13, 147, 148].

6.3. Адаптация к данным. Рассмотренные в настоящем обзоре методы формирования векторных представлений с бинарными и целочисленными компонентами относятся, главным образом, к категории забывчивых методов — они не зависят от конкретных данных, т.е. работают без адаптации к данным (без обучения). Во многих случаях это позволяет получить количественные характеристики ошибок оценки расстояний/сходств.

Однако формирование сохраняющих сходство бинарных векторных представлений (вложений, скетчей, LSH-хэшей) с использованием обучения на имеющихся данных [17, 37, 38, 43, 56, 151, 152] (без учителя — на основе сохранения исходных мер расстояния/сходства данных, или с учителем — на основе информации о сходных и несходных объектах базы) позволяет на практике улучшить (при одинаковой или даже сниженной размерности векторов) результаты в задачах, например, поиска по сходству и классификации.

При этом адаптация к данным затрудняет аналитическое исследование оценок расстояний/сходств по полученным представлениям и характеристик использующих их методов. Кроме того, отличие распределения новых данных от обучающей выборки может ухудшить практические результаты. Для некоторых методов нетривиально формирование представлений объектов, не участвовавших в обучении. Общим недостатком методов с обучением является высокая вычислительная сложность.

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

СПИСОК ЛИТЕРАТУРЫ

1. Rachkovskij D.A. Real-valued embeddings and sketches for fast distance and similarity estimation. *Cybernetics and Systems Analysis*. 2016. Vol. 52, N 6. P.
2. Deza M., Deza E. Encyclopedia of distances. Berlin; Heidelberg: Springer, 2016. 756 p.
3. Lesot M.-J., Rifqi M., Benhadda H. Similarity measures for binary and numerical data: a survey. *Int. J. Knowledge Engineering and Soft Data Paradigm*s. 2009. Vol. 1, N 1. P. 63–84.
4. Choi S.-S., Cha S.-H., Tappert C.C. A survey of binary similarity and distance measures. *J. Systemics, Cybernetics and Informatics*. 2010. Vol. 8, N 1. P. 43–48.
5. Johnson W.B., Lindenstrauss J. Extensions of Lipshitz mapping into Hilbert space. *Contemporary Mathematics*. 1984. Vol. 26. P. 189–206.
6. Indyk P., Motwani R. Approximate nearest neighbors: Towards removing the curse of dimensionality. *Proc. 30th ACM Symp. Theory of Computing*. 1998. P. 604–613.
7. Vempala S.S. The random projection method. Providence, R.I.: American Math. Soc., 2004. 105 p.
8. Matousek J. On variants of the Johnson–Lindenstrauss lemma. *Random Structures and Algorithms*. 2008. Vol. 33, N 2. P. 142–156.
9. Andoni A., Krauthgamer R., Razenshteyn I. P. Sketching and embedding are equivalent for norms. *Proc. STOC'15.* 2015. P. 479–488.
10. Batu T., Ergun F., Sahinalp C. Oblivious string embeddings and edit distance approximations. *Proc. SODA '06*. 2006. P. 792–801.
11. Indyk P., Naor A. Nearest-neighbor-preserving embeddings. *ACM Trans. Algorithms*. 2007. Vol. 3, N 3. Article No 31.
12. Goemans M., Williamson D. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journ. ACM*. 1995. Vol. 42, N 6. P. 1115–1145.
13. Charikar M. Similarity estimation techniques from rounding algorithms. *Proc. STOC'02*. 2002. P. 380–388.
14. Yi X., Caramanis C., Price E. Binary embedding: Fundamental limits and fast algorithm. *JMLR: W&CP.* 2015. Vol. 37. P. 2162–2170.
15. Manku G.S., Jain A., Sarma A.D. Detecting near-duplicates for web crawling. *Proc. WWW'07*. 2007. P. 141–150.
16. Li P., Hastie T.J., Church K.W. Improving random projections using marginal information. *Proc. COLT'06.* 2006. P. 635–649.
17. Yu F.X., Bhaskara A., Kumar S., Gong Y., Chang S.-F. On binary embedding using circulant matrices. arXiv:1511.06480. 5 Dec 2015.

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

19. Rachkovskij D.A. Estimation of vectors similarity by their randomized binary projections. *Cybernetics and Systems Analysis*. 2015. Vol. 51, N 5. P. 808–818.

20. Oehlert G.W. A note on the delta method. *The American Statistician*. 1992. Vol. 46, N 1. P. 27–29.

21. Jacques L., Laska J.N., Boufounos P.T., Baraniuk R. G. Robust 1-Bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Trans. Inf. Theory*. 2013. Vol. 59, N 4. P. 2082–2102.

22. Jacques L. A quantized Johnson–Lindenstrauss lemma: the finding of Buffon's needle. *IEEE Trans. Inf. Theory*. 2015. Vol. 61. N 9. P. 5012–5027.

23. Knuth D.E. Big omicron and big omega and big theta. *ACM Sigact New*s. 1976. Vol. 8. N 2. P. 18–24.

24. Karnin Z, Rabani Y, Shpilka A. Explicit dimension reduction and its applications. *SIAM J. Comput*. 2012. Vol. 41, N 1. P. 219–249.

25. Larsen K. G., Nelson J. Optimality of the Johnson–Lindenstrauss lemma. arXiv:1609.02094. 7 Sep 2016.

26. Plan Y., Vershynin R. Dimension reduction by random hyperplane tessellations. *Discrete and Computational Geometry*. 2014. Vol. 51, N 2. P. 438–461.

27. Oymak S., Recht B. Near optimal bounds for binary embeddings of arbitrary sets. arXiv:1512.04433. 14 Dec 2015.

28. Ailon N., Chazelle B. The Fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput*. 2009. Vol. 39, N 1. P. 302–322.

29. Le Q., Sarlos T., Smola A.J. Fastfood - Computing hilbert space expansions in loglinear time. *JMLR: W&CP*. 2013. Vol. 28, N 3. P. 244–252.

30. Oymak S. Near-optimal sample complexity bounds for circulant binary embedding. arXiv:1603.03178. 14 Mar 2016.

31. Hsieh S.-H., Lu C.-S., Pei S.-C. Fast binary embedding via circulant downsampled matrix: A data-independent approach. *Proc. ICIP'16*. 2016.

32. Choromanska A., Choromanski K., Bojarski M., Jebara T., Kumar S., LeCun Y. Binary embeddings with structured hashed projections. *Proc. ICML'16.* 2016. P. 344–353.

33. Dirksen S., Stollenwerk A. Fast binary embeddings with Gaussian circulant matrices: improved bounds. arXiv:1608.06498. 23 Aug 2016.

34. Li P., Hastie T.J., Church K.W. Very sparse random projections. *Proc. KDD'06*. 2006. P. 287–296.

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

36. Korolev V., Shevtsova I. An improvement of the Berry-Esseen inequality with applications to Poisson and mixed Poisson random sums. *Scandinavian Actuarial Journal*. 2012. Vol. 2012, N 2. P. 81–105.

37. Gong Y., Sanjiv K., Rowley H.A., Lazebnik S. Learning binary codes for highdimensional data using bilinear projections. *Proc. CVPR'13*. 2013. P. 484–491.

38. Zhang X., Yu F.X., Guo R., Kumar S., Wang S., Chang S.-F. Fast orthogonal projection based on kronecker product. *Proc. ICCV'15*. 2015. P. 2929–2937.

39. Indyk P., Motwani R. Approximate nearest neighbors: Towards removing the curse of dimensionality. *Proc. 30th ACM Symp Theory of Computing*. 1998. P. 604–613.

40. Gionis A., Indyk P., Motwani R. Similarity search in high dimensions via hashing. *Proc. VLDB'99*. 1999. P. 518–529.

41. Andoni A., Indyk P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*. 2008. Vol. 51. N 1. P. 117–122.

42. Andoni A. Nearest neighbor search: the old, the new, and the impossible: PhD thesis. Massachusetts Institute of Technology. 2009.

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

44. Li P., Mitzenmacher M., Shrivastava A. Coding for random projections. *Proc. ICML'14*. 2014. P. 676–684.

45. Shalev-Shwartz S., Singer Y., Srebro N. Pegasos: primal estimated sub-gradient solver for SVM. *Proc. ICML'2007*. 2007. P. 807–814.

46. Fan R.-E., Chang K.-W., Hsieh C.-J., Wang X.-R., Lin C.-J. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*. 2008. Vol. 9. P. 1871–1874.

47. Joachims T., Finley T., Yu C.-N. J. Cutting-plane training of structural SVMs. *Machine Learning*. 2009. Vol. 77, N 1. P. 27–59.

48. Martinetz T., Labusch K., Schneegass D. SoftDoubleMaxMinOver: Perceptron-like training of Support Vector Machines. *IEEE Transactions on Neural Networks*. 2009. Vol. 20, N 7. P. 1061–1072.

49. Bottou L. Large-scale machine learning with stochastic gradient descent. *Proc. COMPSTAT'10*. 2010. P. 177–187.

50. Datar M., Immorlica N., Indyk P., Mirrokni V. S. Locality-sensitive hashing scheme based on p-stable distributions. *Proc. SCG'04*. 2004. P. 253–262.

51. Li P., Mitzenmacher M., Shrivastava A. 2-Bit random projections, nonlinear estimators, and approximate near neighbor search. arXiv:1602.06577. 21 Feb 2016.

52. Gorisse D., Cord M., Precioso F. Locality-sensitive hashing for chi2 distance. *IEEE Ttrans. PAMI*. 2012. Vol. 34, N 2. P.402–409.

53. Li P., Samorodnitsky G., Hopcroft J. Sign cauchy projections and chi-square kernel. *Proc. NIPS'13*. 2013. P. 2571–2579.

54. Li P. Sign stable random projections for large-scale learning. arXiv:1504.07235. 27 Apr 2015.

55. Dasgupta A., Kumar R., Sarlos T. Fast locality sensitive hashing. *Proc. SIGKDD'11*. 2011. P. 1073–1081.

56. Pauleve L., Jegou H., Amsaleg L. Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognit. Lett*. 2010. Vol. 31, N 11. P. 1348–1358.

57. Li P. 0-bit consistent weighted sampling. *Proc. KDD'15*. 2015. P. 665-674.

58. Li P. A comparison study of nonlinear kernels. arXiv:1603.06541. 21 Mar 2016.

59. Manasse M., McSherry F., Talwar K. Consistent weighted sampling. Tech. Rep. MSR-TR-2010-73. 2010.

60. Ioffe S. Improved consistent sampling, weighted minhash and L1 sketching. *Proc. ICDM'10*. 2010. P. 246–255.

61. Haeupler B., Manasse M., Talwar K. Consistent weighted sampling made fast, small, and easy. arXiv:1410.4266. 16 Oct 2014.

62. Shrivastava A. Simple and efficient weighted minwise hashing. *Proc. NIPS'16*. 2016.

63. Thorup M. Bottom-k and priority sampling, set similarity and subset sums with minimal independence. *Proc. STOC'1*3. 2013. P. 371–378.

64. Li P. Generalized min-max kernel and generalized consistent weighted sampling. arXiv:1605.05721. 23 May 2016.

65. Li P., Zhang C.-H. Theory of the GMM kernel. arXiv:1608.00550. 1 Aug 2016.

66. Li P. Nystrom method for approximating the GMM kernel. arXiv:1607.03475. 12 Jul 2016.

67. Cristianini N., Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods. Cambridge, UK: Cambridge University Press, 2000. 204 p.

68. Steinwart I., Christmann A. Support Vector Machines. New York: Springer, 2008. 601 p.

69. Hofmann T., Scholkopf B., Smola A. Kernel methods in machine learning. *Annals of Statistics*. 2008. Vol. 36, N 3. P. 1171–1220.

70. Shervashidze N., Schweitzer P., van Leeuwen E.J., Mehlhorn K., Borgwardt K.M. Weisfeiler–Lehman graph kernels. *J. of Machine Learning Research*. 2011. Vol. 2. P. 2539–2561.

71. Luqman M.M., Ramel J.Y., Llados J., Brouard T. Fuzzy multilevel graph embedding. *Pattern Recognition*. 2013. Vol. 46, N 2. P. 551–565.

72. Livi L., Rizzi A., Sadeghian A. Optimized dissimilarity space embedding for labeled graphs. *Information Sciences*. 2014. Vol. 266. P. 47–64.

73. Neumann M., Garnett R., Bauckhage C., Kersting K. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*. 2016. Vol. 102, N 2. P. 209–245.

74. Gartner T., Lloyd J., Flach P. Kernels and distances for structured data. *Machine Learning*. 2004. Vol. 57, N 3. P.205–232.

75. Shin K., Kuboyama T. A generalization of Haussler's convolution kernel Mapping kernel and its application to tree kernels. *J. Comput. Sci. Technol*. 2010. Vol. 25, N 5. P. 1040–1054.

76. Da San Martino G., Navarin N., Sperduti A. A tree-based kernel for graphs. *Proc. ICDM'12*. 2012. P. 975–986.

77. Kriege N., Mutzel P. Subgraph matching kernels for attributed graphs. *Proc. ICML'12*. 2012. P. 1015–1022.

78. Rahimi A., Recht B. Random features for large-scale kernel machines. *Proc. NIPS'07*. 2007. P. 1177–1184.

79. Raginsky M., Lazebnik S. Locality-sensitive binary codes from shift invariant kernels. *Proc. NIPS'09*. 2009. P. 1509– 1517.

80. Kim S., Choi S. Bilinear random projections for locality-sensitive binary codes. *Proc. CVPR'15*. 2015. P. 1338–1346.

81. Kulis B., Grauman K. Kernelized locality-sensitive hashing. *IEEE Trans. PAMI*. 2012. Vol. 34, N 6. P. 1092–1104.

82. Jiang K., Que Q., Kulis B. Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. *Proc. CVPR'15*. 2015. P. 4933–4941.

83. Xia H., Wu P., Hoi S.C., Jin R. Boosting multi-kernel localitysensitive hashing for scalable image retrieval. *Proc. SIGIR'12*. 2012. P. 55–64.

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

85. Li P., König A.C. Theory and applications of b-bit minwise hashing. *Communications of the ACM*. 2011. Vol. 54, N 8. P. 101–109.

86. Kushilevitz E., Ostrovsky R., Rabani Y. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing*. 2000. Vol. 30. N 2. P. 457–474.

87. Li P., Church K.W. A sketch algorithm for estimating two-way and multi-way associations. *Computational Linguistics*. 2007. Vol. 33. N 3. P. 305–354.

88. 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.

89. Flajolet P., Martin G.N. Probabilistic counting algorithms for data base applications. *J. Comput. System Sci.* 1985. Vol. 31. P. 182–209.

90. Cohen E. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. System Sci*. 1997. Vol. 55. P. 441–453.

91. Cohen E. All-distances sketches, revisited: HIP estimators for massive graphs analysis. *Proc. PODS'14*. 2014. P. 88-99.

92. Broder A.Z. On the resemblance and containment of documents. *Proc. SEQUENCES'97*. 1997. P. 21–29.

93. Broder A.Z, Glassman S.C., Manasse M.S., Zweig G. Syntactic clustering of the web. *Computer Networks and ISDN Systems*. 1997. Vol. 29, N 8–13. P. 1157–1166.

94. Broder A.Z., Charikar M., Frieze A.M., Mitzenmacher M. Min-wise independent permutations. *J. Comput. System Sci*. 1998. Vol. 60. P. 327–336.

95. Mitzenmacher M., Pagh R., Pham N. Efficient estimation for high similarities using odd sketches. *Proc. WWW'14*. 2014. P. 109–118.

96. Indyk P. A small approximately min-wise independent family of hash functions. *Journal of Algorithms*. 2001. Vol. 38, N 1. P. 84–90.

97. Patrascu M., Thorup M. On the k-independence required by linear probing and minwise independence. *ACM Trans. Algorithms*. 2016. Vol. 12, N 1. P. 8:1–8:27.

98. Dahlgaard S., Thorup M. Approximately minwise independence with twisted tabulation. *Proc. SWAT'14*. 2014. P. 134–145.

99. Thorup M. Fast and powerful hashing using tabulation. arXiv:1505.01523. 15 Feb 2016.

100. Mitzenmacher M., Vadhan S. Why simple hash functions work: exploiting the entropy in a data stream. *Proc. SODA'08*. 2008. P. 746–755.

101. Li P., Owen A. B., Zhang C.-H. One permutation hashing. *Proc. NIPS'12*. 2012. P. 3122–3130.

102. Charikar M., Chen K., Farach-Colton M. Finding frequent items in data streams. *Proc. ICALP'02*. 2002. P. 693–703.

103. Flajolet P., Fusy É., Gandouet O., Meunier F. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. *Proc. AofA'07*. 2007. P. 127–146.

104. Shrivastava A., Li P. Densifying one permutation hashing via rotation for fast near neighbor search. *Proc. ICML'14*. 2014. P. 557–565.

105. Shrivastava A., Li P. Improved densification of one permutation hashing. *Proc. UAI'14*. 2014. P. 732–741.

106. Dahlgaard S., Knudsen M.B.T., Rotenberg E., Thorup M. Hashing for statistics over k-partitions. *Proc. FOCS'15*. 2015. P. 1292–1310.

107. Valsesia D., Fosson S.M., Ravazzi C., Bianchi T., Magli E. SparseHash: Embedding Jaccard coefficient between supports of signals. *ICME 2016 Workshops*. 2016. P. 1–16.

108. Kussul E.M., Rachkovskij D.A., Baidyk T.N. Associative-projective neural networks: architecture, implementation, applications. *Proc. Neuro-Nimes'91*. 1991. P. 463–476.

109. Rachkovskij D.A., Kussul E.M., Baidyk T.N. Building a world model with structure-sensitive sparse binary distributed representations. *Biologically Inspired Cognitive Architectures*. 2013. Vol. 3. P. 64–86.

110. Kleyko D., Osipov E.,· Rachkovskij D.A. Modification of holographic graph neuron using sparse distributed representations. *Procedia Computer Science*. 2016. Vol. 88. P. 39–45.

111. Kartashov A., Frolov A., Goltsev A., Folk R. Quality and efficiency of retrieval for Willshaw-like autoassociative networks: III. Willshaw–Potts model. *Network: Computation in Neural System*s. 1997. Vol. 8, N 1. P. 71–86.

112. Frolov A.A., Rachkovskij D.A., Husek D. On information characteristics of Willshaw-like auto-associative memory. *Neural Network World*. 2002 Vol. 12, N 2. P. 141–158.

113. Frolov A.A., Husek D., Rachkovskij D.A. Time of searching for similar binary vectors in associative memory. *Cybernetics and Systems Analysis*. 2006. Vol. 42, N 5. P. 615–623.

114. Eshghi K., Kafai M. Support Vector Machines with sparse binary high-dimensional feature vectors. HPE-2016-30. 2016.

115. Амосов Н.М., Байдык Т.Н., Гольцев А.Д., Касаткин А.М., Касаткина Л.М., Куссуль Э.М., Рачковский Д.А. Нейрокомпьютеры и интеллектуальные роботы. Киев: Наук. думка, 1991. 272 с.

116. Kussul E.M., Rachkovskij D.A., Baidyk T.N. On image texture recognition by associative-projective neurocomputer. *Proc. ANNIE'91*. 1991. P. 453-458.

117. Donaldson R., Gupta A, Plan Y., Reimer T. Random mappings designed for commercial search engines. arXiv:1507.05929. 21 Jul 2015.

118. Olshausen B.A., Field D.J. Sparse coding of sensory inputs. *Curr. Opin. Neurobiol*. 2004. Vol. 14. P. 481–487.

119. Ahmad S., Hawkins J. How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons and active dendrites. arXiv:1601.00720. 13 May 2016.

120. 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.

121. Мисуно И.С., Рачковский Д.А., Слипченко С.В., Соколов А.М. Поиск текстовой информации с помощью векторных представлений. *Проблемы программирования*. 2005. № 4. С. 50–59.

122. Shi Q., Petterson J., Dror G., Langford J., Smola A. J., Vishwanathan S.V.N. Hash kernels for structured data. *J. Mach. Learn. Res*. 2009. Vol. 10. P. 2615–2637.

123. Rachkovskij D.A., Slipchenko S.V., Kussul E.M., Baidyk T.N. Sparse binary distributed encoding of scalars. *Journal of Automation and Information Sciences*. 2005. Vol. 37, N 6. P. 12–23.

124. Rachkovskij D.A., Slipchenko S.V., Misuno I.S., Kussul E.M., Baidyk T.N. Sparse binary distributed encoding of numeric vectors. *Journal of Automation and Information Sciences*. 2005. Vol. 37, N 11. P. 47–61.

125. Rachkovskij D.A., Slipchenko S.V., Kussul E.M., Baidyk T.N. A binding procedure for distributed binary data representations. *Cybernetics and Systems Analysis.* 2005. Vol. 41, N 3. P. 319–331.

126. Kussul E.M., Rachkovskij D.A., Wunsch D.C. The random subspace coarse coding scheme for real-valued vectors. *Proc. IJCNN'99*. 1999. P. 450-455.

127. Rachkovskij D.A., Slipchenko S.V., Kussul E.M., Baidyk T.N. Properties of numeric codes for the scheme of random subspaces RSC. *Cybernetics and Systems Analysis*. 2005. Vol. 41, N 4. P. 509–520.

128. Eshghi K., Kafai M. The CRO Kernel: Using concomitant rank order hashes for sparse high dimensional randomised feature maps. *Proc. ICDE'16*. 2016. P. 721–730.

129. Forbus K., Ferguson R., Lovett A., Gentner D. Extending SME to handle large-scale cognitive modeling. *Cognitive Science*. 2016. Vol. 40, N 7. DOI: 10.1111/cogs.12377.

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

131. Rachkovskij D.A. Some approaches to analogical mapping with structure sensitive distributed representations. *J. Experimental and Theoretical Artificial Intelligence*. 2004. Vol. 16, N 3. P. 125–145.

132. Slipchenko S.V., Rachkovskij D.A. Analogical mapping using similarity of binary distributed representations. *Int. J. Information Theories and Applications.* 2009. Vol. 16, N 3. P. 269–290.

133. Jacques L. Small width, low distortions: quasi-isometric embeddings with quantized sub-gaussian random projections. arXiv:1504.06170. 24 Apr 2015.

134. Jacques L., Cambareri V. Time for dithering: fast and quantized random embeddings via the restricted isometry property. arXiv:1607.00816. 4 Jul 2016.

135. Boufounos P.T., Mansour H., Rane S., Vetro A. Dimensionality reduction of visual features for efficient retrieval and classification. *APSIPA Trans. on Signal and Information Processing*. 2016. Vol. 5. e14. P. 1–14.

136. Boufounos P.T., Rane S., Mansour H. Representation and coding of signal geometry. arXiv:1512.07636. 23 Dec 2015.

137. Lv Q., Charikar M., Li K. Image similarity search with compact data structures. *Proc. CIKM'04*. 2004. P. 208–217.

138. Wang Z., Dong W., Josephson W., Lv Q., Charikar M., Li K. Sizing sketches: a rank-based analysis for similarity search. *Proc. SIGMETRICS'07*. 2007. P. 157–168.

139. Dong W., Charikar M., Li K. Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces. *Proc. SIGIR'08*. 2008. P. 123–130.

140. Min K., Yang L., Wright J., Wu L., Hua X.-S., Ma Y. Compact projection: Simple and efficient near neighbor search with practical memory requirements. *Proc. CVPR'10*. 2010. P. 3477–3484.

141. Chávez E., Navarro G., Baeza-Yates R., Marroquín J.L. Searching in metric spaces. *ACM Computing Surveys*. 2001. Vol. 33, N 3. P. 273–321.

142. Zezula P., Amato G., Dohnal V., Batko M. Similarity search: The metric space approach. New York: Springer, 2006. 220 p.

143. Hjaltason G. R., Samet H. Index-driven similarity search in metric spaces. *ACM Transactions on Database Systems.* 2003. Vol. 28, N 4. P. 517-–580.

144. Becker A., Ducas L., Gama N., Laarhoven T. New directions in nearest neighbor searching with applications to lattice sieving. *Proc. SODA'16.* 2016. P. 10-–24.

145. Muja M., Lowe D.G. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans. on PAMI*. 2014. Vol. 36, N 11. P. 2227–2240.

146. Zhang X., Qin J., Wang W., Sun Y., Lu J. Hmsearch: An efficient hamming distance query processing algorithm. *Proc. SSDBM'13*. 2013. P. 19:1–19:12.

147. Norouzi M., Punjani A., Fleet D. J. Fast exact search in Hamming space with multi-index hashing. *IEEE Trans. PAMI*. 2014. Vol. 36, N 6. P. 1107–1119.

148. Song J., Shen H.T., Wang J., Huang Z., Sebe N., Wang J. A distance-computation-free search scheme for binary code databases. *IEEE Trans. Multimedia*. 2016. Vol. 18, N 3. P. 484–495.

149. Pham N., Pagh R. Scalability and total recall with fast CoveringLSH. *Proc. CIKM'16*. 2016.

150. Jiang Z., Xie L., Deng X., Xu W., Wang J. Fast nearest neighbor search in the hamming space. *Proc. MMM'16*. 2016. P. 325–336.

151. Wang J., Liu W., Kumar S., Chang S.-F. Learning to hash for indexing big data: A survey. *Proc. of the IEEE*. 2016. Vol. 104, N 1. P. 34–57.

152. Wang J., Zhang T., Song J., Sebe N., Shen H.T. A survey on learning to hash. arXiv:1606.00185. 1 Jun 2016.

Надійшла до редакції 17.05.2016

Д.А. Рачковський

БІНАРНІ ВЕКТОРИ ДЛЯ ШВИДКОЇ ОЦІНКИ ВІДСТАНЕЙ ТА СХОЖОСТЕЙ

Анотація. Розглянуто методи та алгоритми швидкої оцінки мір відстані/схожості вхідних даних за векторними представленнями з бінарними або цілочисельними компонентами, що отримані з вхідних даних, які є здебільшого векторами великої розмірності з різними мірами відстані (кутова, евклідова та ін.) та схожості (косинус кута, скалярний добуток та ін.). Обговорено методи без навчання, що використовують головним чином випадкові проекції з наступним квантуванням, а також семплювання. Отримані вектори можна застосовувати в алгоритмах пошуку за схожістю, машинного навчання тощо.

Ключові слова: відстань, схожість, вкладення, скетчі, випадкові проекції, семплювання, бінаризація, квантування, лема Джонсона–Лінденштрауса, ядерна схожість, пошук за схожістю, локально-чутливе хешування.

D.A. Rachkovskij

BINARY VECTORS FOR FAST DISTANCE AND SIMILARITY ESTIMATION

Abstract. This review focuses on methods and algorithms for fast estimation of distance/similarity measures of initial data by vector representations with binary or integer components obtained from initial data. The initial data are mainly high-dimensional vectors with various distance measures (angular, Euclidean, etc.) or similarity measures (cosine, inner product, etc.). The discussed methods are without training and use mostly random projection followed by quantization, as well as sampling. The resulting vectors can be used for similarity search, machine learning, and other algorithms.

Keywords: distance, similarity, embeddings, sketches, random projection, sampling, binarization, quantization, Johnson–Lindenstrauss lemma, kernel similarity, similarity search, locality-sensitive hashing.

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.