Алгоритм приближенного поиска ближайшего цифрового массива в иерархически структурированном наборе данных*

М. М. Ланге, С. Н. Ганебных, А. М. Ланге
lange_mm@ccas.ru, sng@ccas.ru, lange_am@mail.ru
ФИЦ «Информатика и управление» РАН, г. Москва, ул. Вавилова, 44/2

Предлагается алгоритм быстрого приближенного поиска в заданном наборе многомерных цифровых массивов ближайшего соседа к предъявляемому массиву. Дефект приближенного поиска определяется отношением разности расстояний от предъявляемого массива до реально найденного и до ближайшего соседа к расстоянию до ближайшего соседа. Алгоритм использует пирамидальные представления массивов с многоуровневым разрешением и стратегию иерархического поиска. При большом линейном размере массивов и большой мощности набора данных получена асимптотическая оценка вычислительного выигрыша алгоритма приближенного поиска относительно алгоритма точного поиска. Для набора изображений рукописных цифр из базы данных MNIST построены экспериментальные оценки среднего дефекта поиска, стандартного отклонения дефектов поиска и вычислительной сложности алгоритма при различных значениях параметра стратегии поиска. Используя полученные оценки, построена зависимость среднего дефекта поиска от вычислительной сложности алгоритма.

Ключевые слова: *многомерный массив; набор данных; ближайший сосед; пирамидальное представление; приближенный поиск; дефект поиска; вычислительная сложность*

DOI: 10.21469/22233792.2.1.01

1 Введение

Задача поиска в заданном наборе векторов евклидова пространства представителя, достаточно близкого к предъявляемому вектору, известна как задача приближенного поиска ближайшего соседа [1–5]. В пространстве фиксированной размерности $d \geqslant 1$ построены алгоритмы, реализующие поиск в наборе из n векторов представителя на расстоянии $D \leqslant (1+\varepsilon)D_{\min}$ от предъявляемого вектора, где $D_{\min}$ — расстояние до ближайшего соседа, а $\varepsilon > 0$ — допустимое отклонение. Алгоритмы с логарифмической сложностью используют древовидные структуры данных и при больших значениях n и фиксированных d и ε имеют вычислительную сложность $O(\log n)$ [1, 4]. Для сравнения алгоритм полного перебора, реализующий поиск ближайшего представителя, имеет сложность $\Theta(dn)$, и при больших значениях n доля сложности алгоритма приближенного поиска относительно сложности переборного алгоритма составляет $O(n^{-1}\log n)$.

Как правило, мультипликативный коэффициент в оценках сложности известных приближенных алгоритмов растет экспоненциально с увеличением d и по степенному закону с уменьшением ε. Явная зависимость сложности от указанных параметров дана в оценке $O(d\lceil 1 + 6d/\varepsilon\rceil^d \log n)$, полученной на решающем BBD-дереве (Balanced Box-Decomposition tree) [4]. Характер зависимости сложности от размерности d и допустимого отклонения ε от ближайшего соседа ограничивает применение такого алгоритма для поиска массивов размерности $d = N^m$ с параметрами $N \geqslant 10$, $m \geqslant 1$ и, в частности, для изображений

*Работа выполнена при частичной финансовой поддержке РФФИ, проекты № 15-01-04671 и № 15-07-07516.

большого размера. На практике известные алгоритмы эффективны в пространстве малой размерности ($d \leqslant 8$) и не обеспечивают быстрого приближенного поиска цифровых массивов типа изображений размера 1024×1024, для которых $d > 10^6$.

В настоящей работе предложен иерархический алгоритм приближенного поиска на множестве многомерных цифровых массивов большого размера ближайшего представителя к предъявляемому массиву. Алгоритм построен в пространстве пирамидальных представлений массивов с многоуровневым разрешением [6]. Такие представления дают описания цифровых массивов в форме деревьев, индекс ветвления которых определяется параметром размерности m [7, 8]. Основой предлагаемого алгоритма является процедура приближенного поиска ближайшего соседа к предъявляемому объекту в многоуровневой сети эталонов, которая разработана для быстрого распознавания образов в пространстве древовидно-структурированных представлений с многоуровневым разрешением [9]. Получена оценка вычислительной сложности алгоритма и проведена его апробация на множестве изображений рукописных цифр [10]. По результатам апробации построены экспериментальные оценки среднего значения и дисперсии величины $(D - D_{\min})/D_{\min}$ (по множеству предъявляемых изображений) при различных значениях параметра алгоритма поиска.

2 Формализация задачи

Рассматривается источник, порождающий множество массивов X^m, в котором любой массив $x^m \in \mathrm{X}^m$ задан m-мерным кубом ($m \geqslant 1$), содержащим N^m элементов из алфавита $A = \{0, 1, \ldots, q-1\}$ ($q \geqslant 2$). Допустимыми считаются массивы, средние значения элементов которых положительны. Предполагается, что $N = 2^L$, где $L \gg 1$ и любой допустимый массив $x^m \in \mathrm{X}^m$ имеет набор описаний

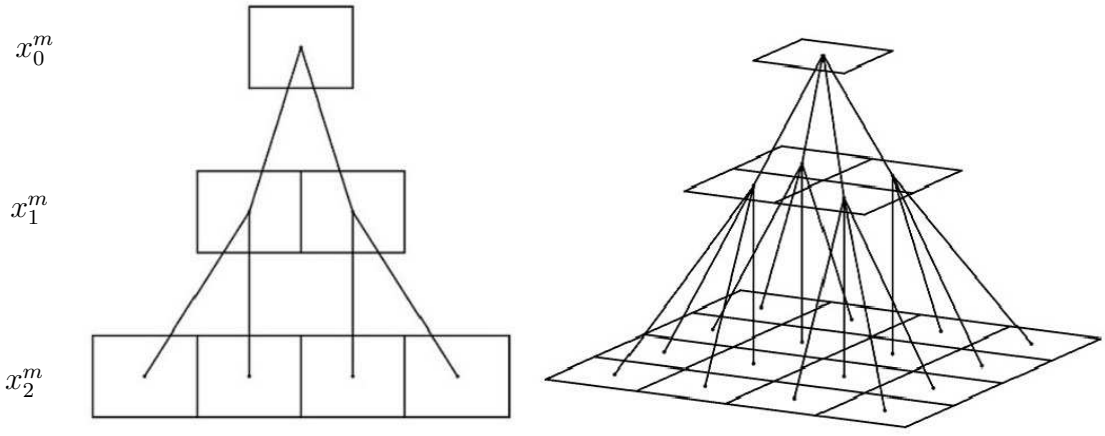
$$\mathbf{x}_L^m = (x_0^m, \ldots, x_l^m, \ldots, x_L^m), \tag{1}$$

образующих 2^m-пирамиду [6] высоты $L = \log_2 N$, в которой описание l-го уровня x_l^m является m-мерным кубом объема 2^{lm}. В случае $m = 1, 2, 3, \ldots$ набор описаний (1) образует соответственно бинарную, квадро- и октопирамиду. Уровни пирамиды строятся рекурсивно: каждый элемент в описании x_l^m вычисляется как среднее значение по 2^m смежным элементам в описании x_{l+1}^m. Основание пирамиды x_L^m совпадает с исходным массивом x^m, вершина x_0^m представлена средним значением элементов массива x^m. Примеры представления одномерного ($m = 1$) и двумерного ($m = 2$) массивов соответственно в форме бинарной пирамиды и квадропирамиды высоты $L = 2$ даны на рис. 1. Бинарная пирамида дает многоуровневое представление последовательности элементов длины N, квадропирамида является многоуровневым представлением изображения размера $N \times N$.

Деление значений элементов в описаниях $x_l^m, l = 1, \ldots, L$, из (1) на значение элемента вершины x_0^m (для допустимых массивов среднее значение элементов больше нуля) дает нормализованное представление

$$\mathbf{y}_L^m = (y_1^m, \ldots, y_l^m, \ldots, y_L^m) \tag{2}$$

в виде последовательности L описаний массива x^m с нарастающим разрешением (числом элементов). Нормализация описаний в представлении (2) обеспечивает их слабую зависимость от размера q используемого алфавита A. Элементы каждого описания y_l^m в представлении (2) снабжены векторами индексов $\mathbf{k}_l^m = (k_{l1}, \ldots, k_{lm})$, где каждый индекс является номером элемента по соответствующей координате m-мерного куба с ребром 2^l и принимает одно из целочисленных значений $1, \ldots, 2^l$. Нормализованные представления (2) образуют множество $\mathbf{Y}_L^m : \mathrm{X}^m \to \mathbf{Y}_L^m$.



(*а*) $m = 1$ (*б*) $m = 2$

Рис. 1 Примеры пирамидальных представлений одномерного и двумерного массивов

Для любой пары m-мерных массивов $x^m \in \mathrm{X}^m$ и $\hat{x}^m \in \mathrm{X}^m$, имеющих нормализованные представления $\mathbf{y}_L^m \in \mathbf{Y}_L^m$ и $\hat{\mathbf{y}}_L^m \in \mathbf{Y}_L^m$ вида (2), описания l-го уровня образованы m-мерными кубами $y_l^m = \{z_{\mathbf{k}_l^m}\} \in \mathbf{y}_L^m$ и $\hat{y}_l^m = \{\hat{z}_{\mathbf{k}_l^m}\} \in \hat{\mathbf{y}}_L^m$, в которых элементы с одинаковыми векторами индексов являются соответственными. Используя соответствие элементов в нормализованных описаниях любой пары массивов, определим для пары массивов $x^m \in \mathrm{X}^m$ и $\hat{x}^m \in \mathrm{X}^m$ меру их различия l-го порядка:

$$D_l(x^m, \hat{x}^m) = 2^{-lm} \sum_{\mathbf{k}_l^m} \left| z_{\mathbf{k}_l^m} - \hat{z}_{\mathbf{k}_l^m} \right| = 2^{-lm} \sum_{k_{l1}=1}^{2^l} \cdots \sum_{k_{lm}=1}^{2^l} \left| z_{k_{l1},\ldots,k_{lm}} - \hat{z}_{k_{l1},\ldots,k_{lm}} \right| . \tag{3}$$

Суммирование мер $D_i(x^m, \hat{x}^m)$, $i = 1, \ldots, l$, вида (3) с весовыми коэффициентами $w_i > 0$ дает взвешенную меру l-го порядка:

$$D_l^w(x^m, \hat{x}^m) = \sum_{i=1}^{l} w_i D_i(x^m, \hat{x}^m) . \tag{4}$$

В качестве весовых коэффициентов в (4) выбираются энтропии уровней пирамиды (2):

$$w_i = \log_2 2^{im} = im. \tag{5}$$

Соотношения (3)–(5) порождают последовательность взвешенных мер различия массивов множества X^m:

$$D_l^w(x^m, \hat{x}^m) , \quad l = 1, \ldots, L , \tag{6}$$

которые определены на множестве нормализованных представлений $\mathbf{Y}_L^m$.

Пусть подмножество массивов $\hat{\mathrm{X}}^m \subset \mathrm{X}^m$ мощности $\|\hat{\mathrm{X}}^m\| = n$ образует набор данных, в котором производится точный или приближенный поиск ближайшего соседа для всех массивов из подмножества $\mathrm{X}^m \setminus \hat{\mathrm{X}}^m$. Для любого предъявляемого массива $x^m \in \mathrm{X}^m \setminus \hat{\mathrm{X}}^m$ решение принимается по мере (6) наибольшего порядка L на наборе $\hat{\mathrm{X}}_*^m \subseteq \hat{\mathrm{X}}^m$, который выбирается согласно принятой стратегии поиска и в общем случае зависит от предъявляемого массива. Решающее правило определяется функцией:

$$\hat{x}_*^m = \arg \min_{\hat{x}^m \in \hat{\mathrm{X}}_*^m} D_L^w(x^m, \hat{x}^m) . \tag{7}$$

Стратегия выбора набора $\hat{\mathrm{X}}^m_*$ в (7) определяет решающий алгоритм, который в случае $\hat{\mathrm{X}}^m_* = \hat{\mathrm{X}}^m$ реализует точный поиск ближайшего массива $\hat{x}^m_* \in \hat{\mathrm{X}}^m$ на основе полного перебора, а в случае $\hat{\mathrm{X}}^m_* \subset \hat{\mathrm{X}}^m$ — приближенный поиск на основе направленного (иерархического) перебора, при котором решение $\hat{x}^m_* \in \hat{\mathrm{X}}^m_*$ совпадает или отличается от ближайшего представителя в $\hat{\mathrm{X}}^m$.

Для выбора набора $\hat{\mathrm{X}}^m_*$ предлагается использовать параметрическую стратегию с параметром $n^* = 1, 2, \ldots, n$, значение которого определяет мощность этого набора $\hat{\mathrm{X}}^m_* : \|\hat{\mathrm{X}}^m_*\| = n^* \leqslant n$. Такая стратегия порождает семейство решающих алгоритмов по критерию (7), включающее алгоритмы приближенного поиска с параметром $n^* < n$ и алгоритм точного поиска с параметром $n^* = n$. Качество алгоритма с параметром n^* на множестве предъявляемых массивов $\mathrm{X}^m \setminus \hat{\mathrm{X}}^m$ определяется средним дефектом поиска

$$\varepsilon_{n^*} = \frac{1}{\|\mathrm{X}^m \setminus \hat{\mathrm{X}}^m\|} \sum_{x^m \in \mathrm{X}^m \setminus \hat{\mathrm{X}}^m} \left(\frac{\min\limits_{\hat{x}^m \in \hat{\mathrm{X}}^m_*} D^w_L(x^m, \hat{x}^m)}{\min\limits_{\hat{x}^m \in \hat{\mathrm{X}}^m} D^w_L(x^m, \hat{x}^m)} - 1 \right) \tag{8}$$

и стандартным отклонением

$$\sigma_{n^*} = \left(\frac{1}{\|\mathrm{X}^m \setminus \hat{\mathrm{X}}^m\|} \sum_{x^m \in \mathrm{X}^m \setminus \hat{\mathrm{X}}^m} \left(\frac{\min\limits_{\hat{x}^m \in \hat{\mathrm{X}}^m_*} D^w_L(x^m, \hat{x}^m)}{\min\limits_{\hat{x}^m \in \hat{\mathrm{X}}^m} D^w_L(x^m, \hat{x}^m)} - 1 \right)^2 - \varepsilon^2_{n^*} \right)^{1/2} \tag{9}$$

дефектов относительно среднего значения (8). Очевидно, что $\varepsilon_{n^*} \geqslant 0$ и $\sigma_{n^*} \geqslant 0$, причем нулевые значения этих статистик достигаются в случае $\hat{\mathrm{X}}^m_* = \hat{\mathrm{X}}^m$ при всех $x^m \in \mathrm{X}^M \setminus \hat{\mathrm{X}}^m$. Вычислительная сложность решающего алгоритма с параметром n^* определяется количеством элементарных операций C_{n^*}, требуемых для поиска решения (7). Рассматривается стратегия выбора набора $\hat{\mathrm{X}}^m_*$ в (7), которая обеспечивает невозрастающие значения ε_{n^*} и σ_{n^*} и неубывающие значения C_{n^*} с ростом n^*.

Решаемая задача заключается в получении оценок характеристик ε_{n^*}, σ_{n^*} и C_{n^*} как функций параметра n^*. Строятся асимптотические оценки вычислительной сложности решающих алгоритмов для источника с параметрами $m \geqslant 1$, $N \to \infty$, $n \to \infty$. Демонстрируется область значений параметра n^*, которая с ростом N обеспечивает стремление к нулю доли сложности алгоритмов приближенного поиска относительно сложности алгоритма точного поиска. Для набора полутоновых изображений рукописных цифр с параметрами $N = 32$, $m = 2$, $n = 50\,000$ строятся экспериментальные оценки ε_{n^*}, σ_{n^*} и C_{n^*} как функции переменной n^*/n на отрезке $[1/n, 1]$. Используя для указанного источника изображений оценки функций ε_{n^*} и C_{n^*} и исключая параметр n^* из условия $C_{n^*} = C^*$ ($C^* > 0$ — заданная допустимая сложность), для заданного набора изображений вычисляется функция «дефект–сложность»:

$$\varepsilon(C^*) = \varepsilon_{n^*} : n^* = \arg(C_{n^*} = C^*)\,. \tag{10}$$

3 Структура набора данных и алгоритм поиска решения

Будем считать, что каждый массив $\hat{x}^m \in \hat{\mathrm{X}}^m$ имеет нормализованное пирамидальное представление $\hat{\mathbf{y}}^m_L$ вида (2). Подмножество таких представлений $\hat{\mathbf{Y}}^m_L : \hat{\mathrm{X}}^m \to \hat{\mathbf{Y}}^m_L$ образует многоуровневую сеть представлений данных

$$\hat{\mathbf{Y}}^m_1, \ldots, \hat{\mathbf{Y}}^m_l, \ldots, \hat{\mathbf{Y}}^m_L, \tag{11}$$

в которой $\hat{\mathbf{Y}}_l^m$ — подмножество представлений $\mathbf{y}_l^m = (y_1^m, \ldots, y_l^m)$, заданных l уровнями нормализованной пирамиды (2). Каждое подмножество в последовательности (11) содержит представления всех массивов из набора $\hat{\mathrm{X}}^m$ и, следовательно, имеет мощность n.

Предлагаемая параметрическая стратегия поиска решения (7) в сети представлений (11) базируется на последовательном сужении зоны поиска на уровнях $l = 1, \ldots, L$ в соответствии с экспоненциальной функцией

$$n_l = \lfloor n 2^{-\alpha m (l-1)} \rfloor, \quad l = 1, \ldots, L\,, \tag{12}$$

с коэффициентом $\alpha = (L-1)^{-1} \log_{2^m}(n/n^*)$, где $n^* = 1, 2, \ldots, n$ — свободный параметр, определяющий мощность набора $\hat{\mathrm{X}}_*^m$ в (7). Значения функции (12) соответствуют количествам массивов, среди которых выполняется поиск на соответствующих уровнях сети (11).

Алгоритм поиска. Для любого предъявляемого массива $x^m \in \mathrm{X}^m$ на последовательных уровнях $l = 1, \ldots, L-1$ сети (11) вычисляются значения взвешенной меры различия $D_l^w(x^m, \hat{x}^m)$ вида (4) по n_l массивам $\hat{x}^m \in \hat{\mathrm{X}}^m$ и среди них отбираются n_{l+1} массивов с наименьшими значениями меры $D_l^w(x^m, \hat{x}^m)$; на уровне $l = L$ среди $n_L = n^*$ массивов отбирается ближайший массив ($n_{L+1} = 1$) с наименьшим значением меры $D_L^w(x^m, \hat{x}^m)$, который дает решение (7).

В случае $1 \leqslant n^* < n$ ($\alpha > 0$) стратегия (12) порождает иерархический алгоритм приближенного поиска ближайшего представителя в наборе данных $\hat{\mathrm{X}}^m$, а в случае $n^* = n$ ($\alpha = 0$) — переборный алгоритм точного поиска. Схемы поиска приближенного и точного решений с помощью указанных алгоритмов даны на рис. 2. В обоих случаях вычисление меры производится рекурсивно с использованием соотношения

$$D_l^w(x^m, \hat{x}^m) = D_{l-1}^w(x^m, \hat{x}^m) + w_l D_l(x^m, \hat{x}^m), \quad l = 1, \ldots, L\,, \tag{13}$$

и начального условия $D_0^w(x^m, \hat{x}^m) = 0$. Мера (13) вычисляется для n_l массивов из набора $\hat{\mathrm{X}}^m$, отбираемых на последовательных уровнях сети (11) согласно (12). В случае приближенного поиска: $n^* \leqslant n_l \leqslant n$, $l = 1, \ldots, L$; в случае точного поиска: $n_l = n$, $l = 1, \ldots, L$.

Вычислительная сложность решающих алгоритмов определяется числом элементарных операций, затрачиваемых на вычисление меры на всех уровнях сети (11), и на сортировку значений меры на последовательных уровнях для отбора ближайших массивов согласно стратегии (12), включая отбор решения на последнем уровне. Элементарной операцией вычисления меры является сравнение пары соответственных элементов в представлениях $\mathbf{y}_l^m$ и $\hat{\mathbf{y}}_l^m$, $l = 1, \ldots, L$, сравниваемых массивов, а элементарной операцией сортировки — сравнение пары значений вычисленной меры на заданном уровне сети (11).

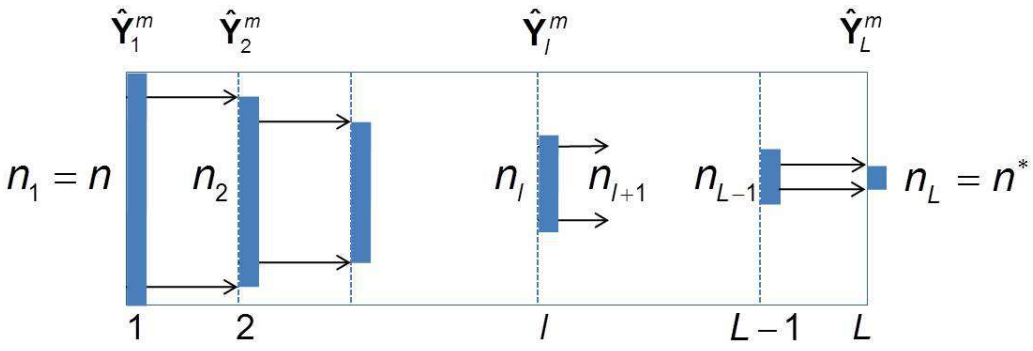


Рис. 2 Схемы поиска приближенного ($n^* < n$) и точного ($n^* = n$) решений

Поскольку число элементов в описаниях l-го уровня равно 2^{ml}, то сложность вычисления меры различия предъявляемого массива с массивами набора данных, отбираемыми согласно стратегии (12), равна

$$C_{n^*}^{\mathrm{msr}} = \sum_{l=1}^{L} n_l 2^{ml} = \sum_{l=1}^{L} \left\lfloor n \left(\frac{n^*}{n} \right)^{(l-1)/(L-1)} \right\rfloor 2^{ml}. \tag{14}$$

Отбор n_{l+1} наименьших значений меры из n_l на уровнях с номерами $l = 1, \ldots, L$ ($n_{L+1} = 1$ соответствует решению) может быть выполнен путем вычисления соответствующей порядковой статистики, что эквивалентно частичной сортировке со сложностью $O(n_l)$ [11]. Поскольку $n_l \leqslant n$ при всех $1 \leqslant l \leqslant L$, то оценка сложности частичной сортировки в решающем алгоритме с параметром $n^* \leqslant n$ имеет вид:

$$C_{n^*}^{\mathrm{srt}} = \begin{cases} O\left(\sum_{l=1}^{L} n_l\right) = O(nL), & n^* < n\,; \\ n-1, & n^* = n\,. \end{cases} \tag{15}$$

В случае $n^* = n$ на уровнях с номерами $l = 1, \ldots, L-1$ отбираются все массивы набора данных, а $(n-1)$ сравнений затрачивается на поиск решения на L-м уровне, что эквивалентно поиску первой порядковой статистики на полном множестве (мощности n) значений меры.

Соотношения (14) и (15) дают асимптотические при $n \to \infty$ оценки вычислительной сложности алгоритма точного поиска с параметром $n^* = n$ ($\alpha = 0$) и алгоритма приближенного поиска с параметром $n^* \leqslant n2^m/N^m$ ($\alpha \geqslant 1$) при $m \geqslant 1$ и $N^m \geqslant \log_q n$. Эти оценки имеют следующий вид:

$$C_{n^* \leqslant n2^m/N^m} = C_{n^* \leqslant n2^m/N^m}^{\mathrm{msr}} + C_{n^* \leqslant n2^m/N^m}^{\mathrm{srt}} \leqslant n2^m L + O(nL) = O(n \log N)\,; \tag{16}$$

$$C_{n^*=n} = C_{n^*=n}^{\mathrm{msr}} + C_{n^*=n}^{\mathrm{srt}} = \frac{2^m}{2^m - 1}(N^m - 1)n + (n-1) = \Omega(nN^m)\,. \tag{17}$$

Из оценок (16) и (17) следует

Утверждение. Доля сложности алгоритма приближенного поиска решения с параметром $n^* \leqslant n2^m/N^m$ относительно сложности алгоритма точного поиска решения с параметром $n^* = n$ удовлетворяет оценке

$$\frac{C_{n^* \leqslant n2^m/N^m}}{C_{n^*=n}} = O\left(\frac{\log N}{N^m}\right)$$

при $m \geqslant 1$, $N^m \geqslant \log_q n$ и $n \to \infty$.

4 Экспериментальные результаты

В данном разделе представлены экспериментальные зависимости показателей качества приближенного поиска ε_{n^*} и σ_{n^*}, определенные соотношениями (8) и (9), и зависимость отношения сложностей $C_{n^*<n}/C_{n^*=n}$ алгоритмов приближенного и точного поиска от величины n/n^*. Указанные зависимости получены для набора полутоновых изображений рукописных цифр из базы данных MNIST [10]. Вычислительный эксперимент выполнен с помощью кода, написанного на языке MATLAB [12]. Параметры изображений: $m = 2$, $N = 32$, $q = 256$; число уровней сети представлений изображений $L = \log_2 N = 5$; мощность

Таблица 1 Оценки качества и сложности поиска

$\log_2(n/n^*)$	$C^{\mathrm{msr}}_{n^*}/C_n$	$C^{\mathrm{srt}}_{n^*}/C_n$	C_{n^*}/C_n	ε_{n^*}	σ_{n^*}
0	0,9993	0,0007	1,0000	0	0
1	0,5325	0,0356	0,5681	0	0
2	0,2885	0,0285	0,3170	0	0
3	0,1597	0,0237	0,1834	0	0
4	0,0908	0,0205	0,1113	2E-6	0,0002
5	0,0535	0,0282	0,0717	3E-6	0,0002
6	0,0329	0,0166	0,0496	4E-6	0.0013
7	0,0214	0,0154	0,0368	8E-6	0,0021
8	0,0146	0,0146	0,0292	0,0004	0,0065
9	0,0107	0,0139	0,0246	0,0007	0,0085

базы $\|\mathrm{X}^m\| = 70\,000$. Цифры на изображениях базы нормированы по размеру и центрированы в поле изображения. В качестве набора данных $\hat{\mathrm{X}}^m$ использован обучающий набор (train set) мощности 60 000; в качестве предъявляемого набора $\mathrm{X}^m \setminus \hat{\mathrm{X}}^m$ — тестовый набор (test set) мощности 10 000. Вычисление характеристик ε_{n^*}, σ_{n^*} и C_{n^*}/C_n выполнено для значений $n/n^* = 2^k$, $k = 0, 1, \ldots, 10$, обеспечивающих коэффициент сужения зоны поиска $\alpha = (\log_{2^m}(n/n^*))/(L-1) = k/(m\log_2(N/2))$ в диапазоне значений $0 \leqslant \alpha \leqslant 5/4$. Следует отметить, что при $k = 0$ ($n^* = n$) алгоритм поиска дает точное решение, а при $k > 0$ ($n^* < n$) — приближенное решение.

Численная оценка вычислительной сложности решающего алгоритма с параметром $n^* < n$ получена с использованием процедуры быстрой сортировки вставками [11], которая на наборе из n элементов имеет среднюю вычислительную сложность $n \log n$. С учетом затрат на вычисление меры и затрат на сортировку значений меры на последовательных уровнях сети представлений данных (11) оценка сложности решающего алгоритма с параметром n^* определяется суммой

$$C_{n^*} = C^{\mathrm{msr}}_{n^*} + C^{\mathrm{srt}}_{n^*}\,, \tag{18}$$

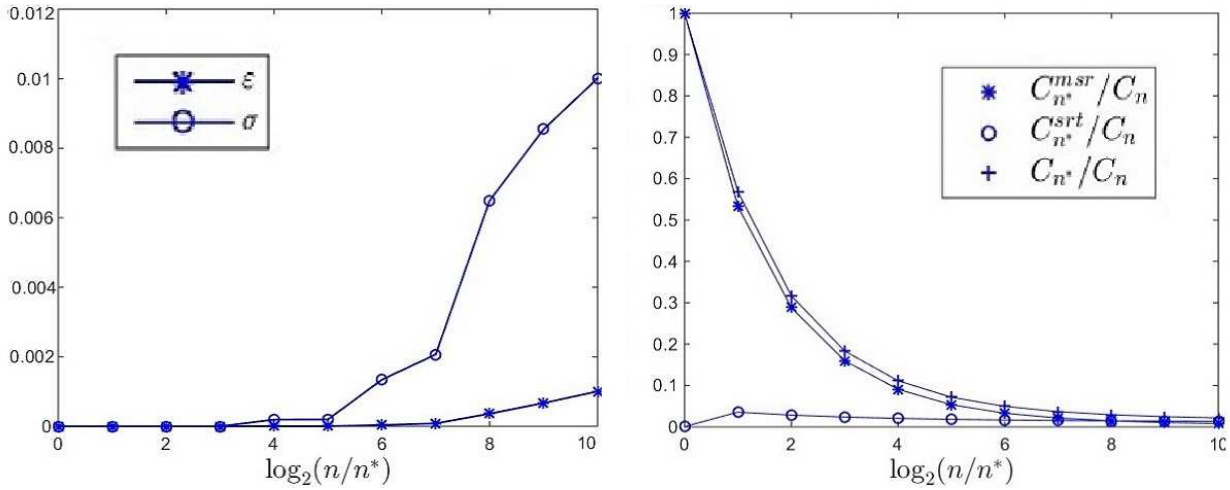
где

$$C^{\mathrm{msr}}_{n^*} = \sum_{l=1}^{L} n_l 2^{ml}\,;\;\; C^{\mathrm{srt}}_{n^*} = (n-1)[n^* = n] + \left((n^*-1) + \sum_{l=1}^{L-1} n_l \log n_l\right)[n^* < n]\,; \tag{19}$$

$[f]$ — индикатор f. В случае $n^* = n$ формулы (18) и (19) дают сложность поиска точного решения, а в случае $n^* < n$ — сложность поиска приближенного решения. Формулы (18) и (19) использованы для вычисления отношения C_{n^*}/C_n при значениях $n/n^* = 2^k$, $k = 0, 1, \ldots, 10$. Экспериментальные оценки характеристик качества поиска ε_{n^*} и σ_{n^*} и численные оценки долей сложности $C^{\mathrm{msr}}_{n^*}/C_n$, $C^{\mathrm{srt}}_{n^*}/C_n$, C_{n^*}/C_n представлены в таблице 1.

Построенные по данным таблицы графики зависимостей ε_{n^*} и σ_{n^*} от $\log_2(n/n^*)$ представлены на рис. 3, *а*, а графики зависимостей $C^{\mathrm{msr}}_{n^*}/C_n$, $C^{\mathrm{srt}}_{n^*}/C_n$ и C_{n^*}/C_n от $\log_2(n/n^*)$ — на рис. 3, *б*. Экспериментальная оценка функции «дефект–сложность» вида (10) представлена графиком зависимости ε_{n^*} от C_{n^*}/C_n на рис. 4.

Графики на рис. 3 демонстрируют вычислительный выигрыш алгоритма приближенного поиска по сравнению с алгоритмом точного поиска от 34,25 до 46,30 раз при сохранении высоких показателей качества приближенного поиска: $0{,}0004 \leqslant \varepsilon_{n^*} \leqslant 0{,}0010$



(*а*) (*б*)

Рис. 3 Характеристики качества (*а*) и сложности (*б*) поиска

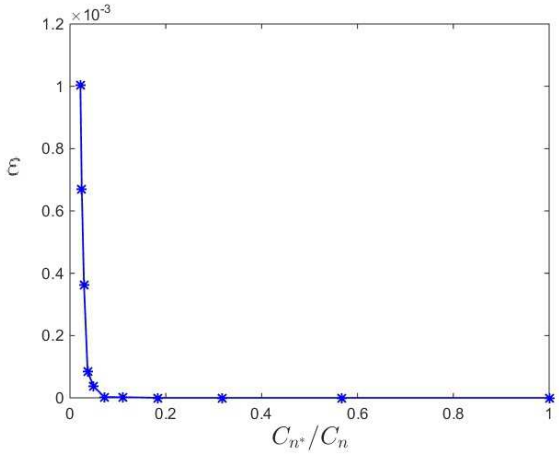


Рис. 4 Функция «дефект–сложность»

и $0{,}0065 \leqslant \sigma_{n^*} \leqslant 0{,}0100$ в диапазоне значений $8 \leqslant \log_2(n/n^*) \leqslant 10$. Показано достижение нулевого дефекта (точного решения) на иерархическом алгоритме, обеспечивающем вычислительный выигрыш в 5,45 раз по сравнению с алгоритмом перебора ($n^* = n/8$). В рамках предложенной стратегии сужения зоны поиска наименьшее значение $C_{n^*=1}/C_n$ (наибольший вычислительный выигрыш) и соответственно наибольший дефект $\varepsilon_{n^*=1}$ дает иерархический алгоритм приближенного поиска с параметром $n^* = 1$.

5 Заключение

Для массивов, заданных m-мерными кубами из N^m элементов дискретного алфавита, предложен иерархический алгоритм приближенного поиска ближайшего соседа к предъявляемому массиву среди множества массивов, образующих набор данных. Разработанный алгоритм ориентирован на ускорение поиска массивов большого размера с параметрами $m \geqslant 1$, $N = 2^L$ при $L \gg 1$, включая изображения с высоким уровнем разрешения. Алгоритм использует пирамидальные представления массивов с многоуровневым разрешением и параметрическую стратегию сужения зоны поиска на последовательных уров-

нях представления набора данных. Показано, что при фиксированной размерности $m \geqslant 1$, большом линейном размере массивов N и большой мощности набора данных n доля вычислительной сложности иерархического алгоритма приближенного поиска относительно сложности переборного алгоритма точного поиска составляет $O(\log N/N^m)$. Экспериментальная апробация разработанного иерархического алгоритма проведена на наборе изображений рукописных цифр из базы данных MNIST. По результатам эксперимента средний дефект иерархического алгоритма приближенного поиска ближайшего соседа оценивается величиной порядка 0,1% при 40-кратном вычислительном выигрыше по сравнению с переборным алгоритмом точного поиска. Дополнительное уменьшение вычислительной сложности приближенного поиска ближайшего соседа может быть достигнуто на объединении алгоритмов, использующих многоуровневое представление массивов и структуру набора данных в форме решающего дерева.

Литература

[1] *Friedman J. H., Bentley J. L., Finkel R. A.* An algorithm for finding best matches in logarithmic expected time // ACM Trans. Math. Softw., 1977. Vol. 3. No. 3. P. 209–226.

[2] *Cleary J. G.* Analysis of an algorithm for finding nearest neighbors in Euclidean space // ACM Trans. Math. Softw., 1979. Vol. 5. No. 2. P. 183–192.

[3] *Soleymani M. R., Morgera S. D.* An efficient nearest neighbor search method // IEEE Trans. Comm., 1987. Vol. 35. No. 6. P. 677–679.

[4] *Arya S., Mount D. M., Netanyahu N. S., Silverman R., Wu A. Y.* An optimal algorithm for approximate nearest neighbor searching in fixed dimensions / J. ACM, 1988. Vol. 45. No. 6. P. 891–923.

[5] *Andoni A., Indyk P.* Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions // Comm. ACM, 2008. Vol. 51. No. 1. P. 117–122.

[6] *Rosenfeld A.* Quadtrees and pyramids for pattern recognition and image analysis // 5th Conference (International) on Pattern Recognition Proceedings. Miami Beach, FL, USA, 1980. P. 802–811.

[7] *Jackins C. L., Tanimoto S. L.* Quadtrees, octtrees, and K-trees: A generalized approach to recursive decomposition of Euclidean space // IEEE Trans. PAMI, 1983. Vol. 5. No. 5. P. 533–539.

[8] *Samet H.* The quadtree and related hierarchical data structures // Computing Survey, 1984. Vol. 16. No. 2. P. 187–260.

[9] *Lange M. M., Stepanov D. Yu.* Recognition of objects given by collections of multichannel images // Pattern Recogn. Image Anal., 2014. Vol. 24. No. 3. P. 431–442.

[10] MNIST database of handwritten digits. `http://www.machinelearning.ru/wiki/index.php?title=MNIST_database_of_handwritten_digits`.

[11] *Cormen T. H., Leiserson C. E., Rivest R. L., Stein C.* Introduction to algorithms. — 3rd ed. — MIT Press, 2009. 1292 p.

[12] Algorithm of approximate search for the nearest neighbour. `https://sourceforge.net/projects/edivis/files/`.

Поступила в редакцию 21.12.2015

Algorithm of approximate search for the nearest digital array in a hierarchical data set*

M. M. Lange, S. N. Ganebnykh, and A. M. Lange
lange_mm@ccas.ru, sng@ccas.ru, lange_am@mail.ru
Federal Research Center "Computer Science and Control" of RAS, 44/2 Vavilova st., Moscow, Russia

An algorithm of approximate fast search in a given set of multidimensional digital arrays for the nearest neighbor of a submitted array is suggested. A search error is defined by a ratio of a difference of distances from a submitted array to the really found array and to the nearest neighbor relative to the distance to the nearest neighbor. The proposed algorithm uses pyramid-based multiresolution representations of the arrays and a hierarchical search strategy. For a large linear size of the arrays and a large cardinality of the data set, an asymptotic computational gain of the approximate search algorithm with respect to the exact search algorithm is estimated. Given data set of grayscale handwritten digit images taken from the MNIST database, a mean search error, a standard deviation of the search errors, and a computational complexity of the algorithm as the appropriate functions of the search parameter are experimentally estimated. Using these estimates, a dependence of the mean search error on the computational complexity is calculated.

Keywords: *multidimensional array; data set; nearest neighbor; pyramid-based representation; approximate nearest search; search error; computational complexity*

DOI: 10.21469/22233792.2.1.01

References

[1] Friedman, J. H., J. L. Bentley, and R. A. Finkel. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.* 3(3):209–226.

[2] Cleary, J. G. 1979. Analysis of an algorithm for finding nearest neighbors in Euclidean space. *ACM Trans. Math. Softw.* 5(2):183–192.

[3] Soleymani, M. R., and S. D. Morgera. 1987. An efficient nearest neighbor search method. *IEEE Trans. Comm.* 35(6):677–679.

[4] Arya, S., D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. 1988. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. ACM* 45(6):891–923.

[5] Andoni A., and P. Indyk. 2008. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Comm. ACM* 51(1):117–122.

[6] Rosenfeld, A. 1980. Quadtrees and pyramids for pattern recognition and image analysis. *5th Conference (International) on Pattern Recognition Proceedings*. Miami Beach, FL. 802–811.

[7] Jackins, C. L., and S. L. Tanimoto. 1983. Quadtrees, octtrees, and K-trees: A generalized approach to recursive decomposition of Euclidean space. *IEEE Trans. PAMI* 5(5):533–539.

[8] Samet, H. 1984. The quadtree and related hierarchical data structures. *Computing Survey* 16(2):187–260.

[9] Lange, M. M., and D. Yu. Stepanov. 2014. Recognition of objects given by collections of multichannel images. *Pattern Recogn. Image Anal.* 24(3):431–442.

[10] MNIST database of handwritten digits. Available at: http://www.machinelearning.ru/wiki/index.php?title=MNIST_database_of_handwritten_digits (accessed February 8, 2016).

*The work was partially supported by the Russian Foundation for Basic Research (grants 15-01-04671 and 15-07-07516).

[11] Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein. 2009. *Introduction to algorithms*. 3rd ed. MIT Press. 1292 p.

[12] Algorithm of approximate search for the nearest neighbour. Available at: `https://sourceforge.net/projects/edivis/files/` (accessed February 8, 2016).

Received December 21, 2015