Сибирский журнал
индустриальной математики
Октябрь–декабрь, 2006. Том IX, № 4(28)

УДК 519.95

СЕМЕЙСТВО АЛГОРИТМОВ ЛОКАТОР ДЛЯ БЫСТРОГО ПОИСКА БЛИЖАЙШЕГО АНАЛОГА*)

Н. Г. Загоруйко, В. В. Дюбанов

Описываются алгоритмы семейства ЛОКАТОР для быстрого направленного поиска ближайшего эталона в процессе распознавания образов. Алгоритмы основаны на пошаговом сокращении количества конкурирующих образов и фокусировании внимания на тех образах, которые имеют наибольшие шансы стать победителями в этой конкуренции. Приведены оценки трудоемкости алгоритмов. Показано, что они слабо зависят от количества образов и размерности признакового пространства.

1. Классификация задач поиска аналогов

При распознавании контрольного объекта решение о принадлежности к одному из образов обычно принимается в результате его сравнения с эталонами всех распознаваемых образов. В работе [1] предложены алгоритмы поиска ближайшего эталона, которые существенно ускоряют процесс принятия решений и мало зависят от количества распознаваемых образов. Алгоритмы типа ЛОКАТОР основаны на пошаговом сокращении количества конкурирующих образов и фокусировании внимания на тех образах, которые имеют наибольшие шансы оказаться ближайшим аналогом распознаваемого объекта. В данной работе представлены результаты более детального исследования алгоритмов такого рода.

Мерой «аналогичности» между контрольным объектом Z и эталоном i-го образа S_i служит величина, обратно пропорциональная расстоянию $R(z,i)$ между ними в пространстве их описания. Содержание задач поиска аналога и методов их решения зависит от ответа на следующие вопросы.

1. Описываются образы значениями признаков или парными расстояниями между всеми образами? Если признаками, то мы имеем дело с обычной задачей распознавания в N-мерном пространстве X. Если K образов описаны только значениями парных расстояний между ними (т. е. представлены матрицей M расстояний r размером $K \times K$), то мы решаем задачу распознавания в «беспризнаковом» пространстве образов Y.

2. Если решается задача распознавания в признаковом пространстве X, то какова метрика этого пространства? Будем рассматривать три вида метрик. В метрике L_∞ расстояние между объектами a и b равно максимальному расстоянию из всех покоординатных расстояний между ними:

$$R_\infty(a,b) = \max r_j(a,b), \quad j = 1, \ldots, N.$$

В метрике L_2 (евклидовой) расстояния равны квадратному корню из суммы квадратов расстояний по координатам:

$$R_2(a,b) = \sqrt{\sum_j^N r^2(a,b)}.$$

*) Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 05–01–00241а).

В метрике L_1 расстояние R между объектами a и b определяется суммой расстояний между ними по всем координатам:

$$R_1(a,b) = \sum_j^N r_j(a,b).$$

3. Задается или не задается порог d допустимых различий между объектом Z и эталоном образа-аналога? Если порог d задается, то мы имеем дело с поиском d-аналога. Если порогового условия нет, то ищется ближайший abs-аналог вне зависимости от того, на каком расстоянии от Z он находится.

4. Требуется найти все d-аналоги или один самый близкий? Если один, то выбирается ближайший abs-аналог или ближайший d-аналог. Если все, то выбираются $k < K$ образов, являющихся d-аналогами объекта Z.

Сочетания разных ответов на эти вопросы порождают 12 различных задач поиска аналога:

1) $(L_\infty, d, -)$: найти все d-аналоги в пространстве X с метрикой L_∞;

2) $(L_\infty, d, 1)$: найти один ближайший d-аналог в пространстве X с метрикой L_∞;

3) $(L_\infty, -, 1)$: найти один ближайший abs-аналог в пространстве X с метрикой L_∞.

Аналогична интерпретация и остальных задач в пространстве X:

4) $(L2, d, -)$; 5) $(L2, d, 1)$; 6) $(L2, -, 1)$; 7) $(L1, d, -)$; 8) $(L1, d, 1)$; 9) $(L1, -, 1)$.

В пространстве Y можно использовать лишь парные расстояния r между объектами, что порождает следующие задачи:

10) $(r, d, -)$; 11) $(r, d, 1)$; 12) $(r, -, 1)$.

2. Поиск аналогов в метрике L_∞

2.1. Задача $(L_\infty, d, -)$**.** Поиск всех аналогов, которые удовлетворяют условию $R_\infty \le d$, можно ускорить путем рассмотрения проекций точки Z и эталонов K образов на отдельные координаты. Пусть пороговое условие определяет максимально допустимые расстояния d между объектом Z и эталоном S_i по каждой t-й характеристике. Тогда можно считать, что если хотя бы по одной координате это расстояние больше d, то i-й образ из списка конкурентов на роль ближайшего аналога можно вычеркнуть. Для оставшихся $k < K$ образов та же процедура повторяется с использованием проекции на вторую координату и т. д. На этом основан алгоритм ЛОКАТОР-1 $(L_\infty, d, -)$, который состоит из следующих шагов.

Шаг 0. $t = 0$.

Шаг 1. $t = t + 1$. Выбирается координата X_t.

Шаг 2. Вычисляются расстояния R_t от проекции точки Z до проекций всех образов S_i, $i = 1, 2, \ldots, K$, на эту координату: $R_t(z, i)$.

Шаг 3. В списке претендентов остается $k < K$ образов, для которых $R_t(z, i) \le d$.

Шаг 4. Если $k = 0$, то принимается решение о том, что среди K образов нет образа, который можно считать d-аналогом объекта Z. Алгоритм останавливается.

Шаг 5. Если $k > 0$ и $t < N$, то происходит возврат на шаг 1.

Шаг 6. Если после шага $t = N$ остается $k > 1$ образов, то все они являются d-аналогами объекта Z.

Для проведения исследований этого и всех других алгоритмов семейства ЛОКАТОР была разработана программа LOCATEST. Ее генератор данных позволяет создать массив из K N-мерных векторов, распределенных по равномерному и нормальному закону с разной дисперсией D. Рабочие параметры можно менять в широком диапазоне значений:

количество K распознаваемых образов — от 2 до 5000;

размерность N признакового пространства X — от 1 до 500;

дисперсия D распределения эталонов образов в пространстве X — от 0,1 до 500;

допустимый порог d «неаналогичности», который определяется отношением $d = R(z,i)/D$, — от 0 до 1;

количество повторов G генерации выборки — от 1 до 1000;

предельное число H шагов поиска порога — от 1 до 100;

предельное число Q используемых локаторов — от 1 до 5000;

количество T контрольных объектов — от 1 до 5000.

В процессе экспериментов выяснилось, что если при заданном пороге d процесс останавливается на шаге $t = N$, то доля претендентов (в % от K) по мере увеличения числа рассмотренных признаков (от шага $t = 1$ к шагу $t = N$) сокращается приблизительно по формуле $k = Ke^{-0,6t}$ и не зависит от дисперсии D и числа образов K.

Если трудоемкость алгоритма сравнения распознаваемого объекта со всеми K образами равна $P_0 = cKN$, то трудоемкость алгоритма ЛОКАТОР-1 при этом условии равна $P_1 = 0,16cKN$. Следовательно, данный алгоритм сокращает время поиска ближайшего аналога приблизительно в шесть раз. Но следует иметь в виду, что при малых значениях d процесс может заканчиваться и на шаге $t < N$, так что приведенная оценка трудоемкости P_1 является оценкой сверху.

2.2. Задача $(L_\infty, d, 1)$**.** Поиск одного ближайшего аналога, который удовлетворяет условию $R_\infty \le d$, делается с помощью алгоритма ЛОКАТОР-2 $(L_\infty, d, 1)$.

Шаг 0. $t = 0$.

Шаг 1. $t = t + 1$. Выбирается координата X_t.

Шаг 2. Вычисляется расстояние от проекции точки Z до проекций всех образов S_i, $i = 1, 2, \ldots, K$, на эту координату: $R_t(z,i)$.

Шаг 3. В списке претендентов остается $k < K$ образов, для которых $R_t(z,i) \le d$.

Шаг 4. Если $k = 0$, то принимается решение о том, что среди K образов нет образа, который можно считать d-аналогом объекта Z. Алгоритм останавливается.

Шаг 5. Если $k > 0$ и $t < N$, то происходит возврат на шаг 1.

Шаг 6. Если $t = N$ и $k = 1$, то образ, оставшийся невычеркнутым, объявляется искомым ближайшим d-аналогом.

Шаг 7. Если $t = N$ и $k > 1$, то методом последовательных приближений для оставшихся k образов подбирается такой порог $d_0 < d$, при котором на шаге $t = N$ получается $k = 1$.

Исследования этого алгоритма проведены методом имитационного моделирования с помощью программы LOKATEST. При каждом сочетании N и D методом последовательных приближений были найдены значения оптимальных порогов d_0, с которых нужно начинать, чтобы на шаге $t = N$ обычно оставался один конкурент на роль ближайшего аналога. Полученные значения начальных порогов для дисперсий от 1 до 500 достаточно точно приближаются следующей формулой:

$$d_0 = (D/500)(0,04(\log_{10} N)^2 + 0,21 \log_{10} N).$$

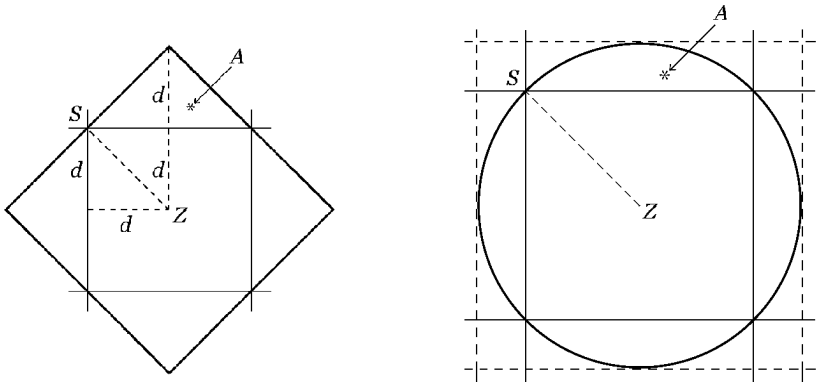
Если заданный порог $d < d_0$, то факт отсутствия аналога при заданном пороге будет обнаруживаться уже на первых шагах алгоритма. Так, при $d < 0,98d_0$ почти всегда получается ответ «аналог отсутствует». Если же порог $d < 0,9d_0$, то этот ответ можно выдавать, не запуская программу.

Процедуры подбора порога на шаге 7 требуют дополнительного расхода времени. Однако основная часть из них оперирует небольшим числом образов. В итоге алгоритм ЛОКАТОР-2 $(L_\infty, d, 1)$, использующий режим подбора порога, приблизительно в 4 раза эффективнее метода сравнения объекта Z со всеми образами по всем признакам.

2.3. Задача $(L_\infty, -, 1)$**.** В метрике L_∞ ближайший d-аналог одновременно является и abs-аналогом. Поэтому данная задача решается алгоритмом ЛОКАТОР-2 $(L_\infty, d, 1)$.

3. Поиск аналогов в метриках L_1 и L_2

В метриках L_1 и L_2 ближайший аналог, выбранный по координатному порогу d, не будет являться одновременно и ближайшим аналогом без порога (abs-аналогом).



а *б*

Рис. 1. Эквидистантные линии

В пространстве L_1 точки, удаленные от точки Z на расстояния, равные $R(z,s) = d + d$, лежат на поверхности гиперкуба с длиной ребра, равной $2d\sqrt{N}$ (жирные линии на рис. $1, a$). Точка A ближе к точке Z, чем точка S, но она будет отсечена порогом d. Ситуация в метрике L_2 аналогична: точки, равноудаленные от точки Z, лежат на поверхности гиперсферы диаметром $2d\sqrt{N}$ (см. рис. $1, b$). Точка A ближе к Z, чем точка S, но пороговое условие d вычеркнет ее из числа претендентов на роль ближайшего abs-аналога.

3.1. Задача $(L_2, d, -)$**.** Если требуется найти все d-аналоги в пространстве L_2 (евклидовом), то следует пользоваться алгоритмом ЛОКАТОР-3 $(L_2, d, -)$, который состоит в следующем.

Шаг 0. $t = 0$.

Шаг 1. $t = t + 1$. Если $t < N$, то выбирается координата X_t.

Шаг 2. Вычисляются расстояния $R_t(z, j)$ между проекциями на ось X_t объекта Z и всех образов S_j, $j = 1, \ldots, k_t$, которые остались в списке претендентов к t-му шагу.

Шаг 3. Эти расстояния добавляются в счетчики евклидовых расстояний R_j для соответствующих образов S_j, $j = 1, \ldots, k_t$.

Шаг 4. В списке претендентов остается k образов, для которых $R_j \le d$. Если $k = 0$, то выдается решение «d-аналогов нет».

Шаг 5. Если $t < N$ и $k > 0$, то происходит возврат к шагу 1.

Шаг 6. Если $t = N$ и $k > 0$, то все оставшиеся образы являются d-аналогами Z.

3.2. Задача $(L_2, d, 1)$**.** Если нужно найти один самый близкий d-аналог, то следует применить алгоритм ЛОКАТОР-4 $(L_2, d, 1)$, который состоит в следующем.

Шаг 0. $t = 0$.

Шаг 1. $t = t + 1$. Выбирается координата X_t.

Шаг 2. Вычисляются расстояния $R_t(z,j)$ между проекциями на ось X_t объекта Z и всех образов S_j, $j=1,\ldots,k_t$, которые остались в списке претендентов к t-му шагу.

Шаг 3. Эти расстояния добавляются в счетчики евклидовых расстояний R_j для соответствующих образов S_j, $j=1,\ldots,k_t$. Вычеркиваются образы, для которых $R_j > d$.

Шаг 4. Среди оставшихся k_t образов определяется образ S^*, для которого $R_j = R_{\min}$. На этом этапе он является сильнейшим претендентом на роль ближайшего d-аналога объекту Z.

Шаг 5. Для всех образов S_j, $j=1,\ldots,k_t$, определяется разность $F_j = R_j - R_{\min}$.

Шаг 6. Если $F_j > R_{\min} + R^*_{N-t}$, то образ S_j не сможет выиграть соревнование у образа S^*, даже если по каждому из оставшихся $N-t$ признаков он не отличим от Z, а образ S^* по каждому из этих признаков будет удален от Z на максимально возможное расстояние R^*. После вычеркивания таких образов S_j в списке претендентов остается k_t образов.

Шаг 7. Если число конкурентов $k_t > 1$ и $t < N$, то происходит возврат к шагу 1.

Шаг 8. Если $k_t = 0$, то выдается сообщение «d-аналогов нет».

Шаг 9. Если $k_t = 1$, то он является абсолютным ближайшим аналогом объекта Z.

Шаг 10. Если $k_t > 1$, то все они объявляются *abs*-аналогами объекта Z.

3.3. Задача $(L_2,-,1)$**.** Состоит в нахождении одного образа, который является ближайшим беспороговым *abs*-аналогом объекта Z. Задача решается алгоритмом ЛОКАТОР-5 $(L_2,-,1)$, который отличается от предыдущего алгоритма ЛОКАТОР-4 $(L_2,d,1)$ только тем, что из шага 3 исключается проверка по условию $R_j > d$.

Эксперименты показали, что при малых дисперсиях D условие F_j выполняется только на последних шагах. Это объясняется тем, что расстояния $R(z,j)$ от объекта Z до всех образов мало отличаются друг от друга. При большой дисперсии вычеркивание начинается несколько раньше. Но в целом алгоритм ЛОКАТОР-5 почти не ускоряет процесс беспорогового поиска ближайшего аналога по сравнению с методом всех сравнений.

На первых шагах алгоритма, когда $N-t$ велико, вероятность того, что на всех оставшихся $N-t$ признаках расстояние от Z до S_j будет равно единице, мала. На этом основании подбором значения $R^* < 1$ в зависимости от $N-t$ можно существенно ускорить работу алгоритма и обеспечить правильное решение в подавляющем большинстве случаев. Но гарантировать точное решение при таком подходе нельзя. Так что этот вероятностный подход можно использовать, если требуется найти k образов, среди которых ближайший *abs*-аналог находится с заданной вероятностью P.

3.4. Задачи $(L_1,d,-)$**,** $(L_1,d,1)$**,** $(L_1,-,1)$**.** Эти три задачи отличаются от трех предыдущих только тем, что расстояния между Z и эталонами образов считаются по правилам метрики L_1. Алгоритмы для решения данных задач те же, что используются для решения трех предыдущих задач. В них заменяется только процедура оценки расстояний R_j.

4. Беспризнаковое пространство Y

Задачи 10, 11 и 12 решаются в беспризнаковом пространстве образов Y. Оно может быть задано изначально, например, в случае, когда эксперты не могут в явном виде указать отдельные характеристики объектов, но могут оценить меру различия между любой парой объектов. Результаты такого экспертного оценивания имеют вид матрицы размерности $K \times K$.

К аналогичному виду данных можно перейти и в том случае, когда объекты описаны признаками метрического пространства X. С переходом от пространства X к пространству Y связана надежда на справедливость гипотезы о том, что парные расстояния между образами позволят ускорить поиск ближайшего аналога. Эту гипотеза проверялась в ходе данного исследования.

Каждая i-я строка матрицы M, имеющей размерность $K \times K$, содержит информацию о том, на каком расстоянии $R(i,j)$ от образа S_i находятся все остальные j-е образы. Сформируем пространство из K ортогональных координат. Если образ S_i совместить с началом координат и на оси Y_i отметить точки, отстоящие от начала координат на расстоянии $R(i,j)$, то можно считать эти точки проекциями всех j-х образов на ось Y_i. Аналогичным путем можно построить проекции эталонов всех образов на все K координатных осей. В результате каждый образ будет представлен в этом пространстве K-мерным вектором, и само это пространство Y имеет свойства обычного векторного пространства над полем действительных чисел. В нем выполняются все аксиомы метрического пространства, включая свойства треугольника.

Связь исходного признакового пространства X с этим «пространством образов» Y однозначна в одну сторону: от X к Y. Обратный переход, который рассматривается в статистических задачах многомерного шкалирования, не обеспечивает однозначного восстановления исходного признакового пространства. Отношения мер близости между несколькими образами в пространствах X и Y не одинаковы. Пространство Y характеризует не свойства объектов, а структуру взаимного расположения объектов друг относительно друга в пространстве X, их «структурные роли» в среде этих объектов. Наличие матрицы M со структурной информацией позволяет строить направленные процедуры поиска ближайшего аналога.

4.1. Задача $(r,d,-)$**.** Задача поиска всех образов, удаленных от Z не более чем на расстояние d, решается алгоритмом ЛОКАТОР-6 $(r,d,-)$. При этом требуется задать порог d и предельное значение количества локаторов Q.

Шаг 0. Задается порог d и предельное число локаторов $Q < K$, которые можно использовать для нахождения аналога. Устанавливается счетчик числа использованных локаторов $q = 1$.

Шаг 1. Определяется расстояние $R(z,i)$ от объекта Z до случайно выбранного первого образа-локатора S_i.

Шаг 2. По i-й строке матрицы M находятся образы, расстояния которых от образа S_i лежат вне диапазона $R(z,i) \pm d$, и эти образы исключаются из дальнейшего рассмотрения. В списке конкурентов образу S_i остается k образов.

Шаг 3. Если $k = 0$ и $R(z,i) > d$, то ближайшего аналога объекту Z в базе нет (принимается решение, что Z принадлежит новому $(K+1)$-му образу).

Шаг 4. Если $k > 1$ и $q < Q$, то $q = q+1$ и по матрице M выбирается такой следующий образ-локатор S_v, при котором величина $F = |R(z,i) - R(v,i)|$ минимальна. С его участием повторяются процедуры, аналогичные шагам 1–3: определение расстояния $R(z,v)$, сокращение списка конкурентов и проверка остающихся конкурентов.

Шаг 5. Если число q использованных локаторов стало равным Q и при этом $k > 1$, то все оставшиеся невычеркнутыми образы объявляются d-аналогами объекта Z.

Было проведено исследование влияния порядка выбора локаторов на эффективность алгоритма. В модельных экспериментах обнаружено, что процедура направленного выбора локаторов по сравнению со случайным выбором очередного локатора уменьшает количество используемых локаторов на 10–20 %. Вместе с тем на направленный выбор затрачивается дополнительное машинное время. Решение вопроса о том, использовать ли режим направленного перебора, зависит от сравнения затрат F_1 на направленный поиск очередного локатора с затратами F_2 на оценивание расстояния от Z до одного локатора. Если $F_2 > F_1$,

то замедление процесса, вызванное направленным поиском очередного локатора, в итоге может оказаться вполне оправданным.

4.2. Задача $(r, d, 1)$**.** Задача поиска ближайшего аналога, удаленного от объекта Z не более, чем на расстояние d, решается алгоритмом ЛОКАТОР-7 $(r, d, 1)$.

Шаг 0. Задается предельное число локаторов $Q < K$ и максимальное число H шагов приближения порога, которые можно использовать для нахождения аналога. Устанавливается счетчик числа использованных локаторов $q = 1$ и шагов $h = 0$.

Шаг 1. Определяется расстояние $R(z, i)$ от объекта Z до случайно выбранного первого образа-локатора S_i.

Шаг 2. По i-й строке матрицы M находятся образы, расстояния которых от образа S_i лежат вне диапазона $R(z, i) \pm d$, и эти образы исключаются из дальнейшего рассмотрения. В списке конкурентов образу S_i остается k образов.

Шаг 3. Если $k = 0$ и $R(z, i) \leq d$, то образ S_i объявляется ближайшим аналогом контрольного объекта Z.

Шаг 4. Если $k = 0$ и $R(z, i) > d$, то ближайшего аналога объекту Z в базе нет (принимается решение, что Z принадлежит новому $(K + 1)$-му образу).

Шаг 5. Если $k = 1$, то анализируется расстояние $R(z, j)$ от Z до этого единственного конкурента S_j. Из двух образов S_i и S_j аналогом объекта Z считается образ, ближайший к Z и удаленный от него на расстояние $R(z) < d$. Если это условие для S_j не выполняется, то принимается решение о том, что ближайшим аналогом является образ S_i.

Шаг 6. Если $k > 1$ и $q < Q$, то $q = q + 1$ и по матрице M выбирается такой следующий образ-локатор S_v, при котором величина $F = |R(z, i) - R(v, i)| + |R(z, j) - R(v, j)|$ минимальна. С его участием повторяются процедуры, аналогичные шагам 1–5: определение расстояния $R(z, v)$, сокращение списка конкурентов и проверка остающихся конкурентов.

Шаг 7. Если число q использованных локаторов стало равным Q и при этом $k > 1$, то методом последовательных приближений за число шагов $h \leq H$ выбирается такой порог $d^* < d$, при котором остается один образ (или несколько образов, неразличимых по расстоянию $R(x)$).

Шаг 8. Алгоритм останавливается, если обнаруживается, что $k = 1$ или $h = H$. Этот единственный образ или несколько оставшихся невычеркнутыми объявляются *abs*-аналогами объекта Z.

Для ускорения этого поиска путем имитационного моделирования различных ситуаций были выбраны оптимальные пороги d_0, при которых обычно за небольшое число шагов t удается найти наиболее близкий аналог объекту Z. Эти оптимальные начальные значения порога d_0 хорошо приближаются следующей формулой:

$$d_0 = (D/500)(0,03(\log_{10} N)^2 + 0,15 \log_{10} N).$$

Алгоритм ЛОКАТОР-7 $(r, d, 1)$ вне зависимости от параметров N, D и K затрачивает на процедуру поиска ближайшего аналога приблизительно одинаковый ресурс, пропорциональный величине qV, где V — ресурс, необходимый для оценки расстояния от объекта Z до одного локатора. По сравнению с методом сравнения Z со всеми K образами это упрощает поиск в K/q раз. При $K > 10000$ ускорение поиска превышает три порядка.

Программа, реализующая этот алгоритм, так же, как и предыдущая, имеет два режима выбора следующего локатора (направленный и случайный).

4.3. Задача $(r, -, 1)$**.** Задача распознавания одного ближайшего *abs*-аналога в пространстве образов Y решается алгоритмом ЛОКАТОР-8 $(r, -, 1)$. Как и в предыдущем случае, распознавание объекта Z начинается с оценки расстояния $R(z, i)$ до эталона любого образа S_i. Если $R(z, i) = 0$, то процесс распознавания завершен. В противном случае нужно исключить из рассмотрения те

образы, к которым этот объект принадлежать заведомо не может. Для этого воспользуемся следующим легко доказываемым утверждением.

Утверждение. *Эталоны всех образов, удаленных от эталона S_i на расстояние $R > 2R(z,i)$, находятся по отношению к объекту Z дальше, чем эталон S_i.*

В результате список рассматриваемых образов может быть сокращен до $k_1 < K$ образов. Для направленного выбора следующего локатора по i-й строке матрицы M находим эталон S_j, для которого величина $F = |R(z,i) - R(j,i)|$, $j = 1, \ldots, k_1$, минимальна. Оценивается расстояние $R(z,j)$ от объекта Z до эталона S_j. Знание расстояний от точки Z до двух эталонов дает возможность более точно пеленговать позицию точки Z по отношению ко всем оставшимся образам. По критерию $R > 2R(z,j)$ список рассматриваемых конкурентов сокращается до величины $k_2 \le k_1$ и среди них в качестве третьего претендента выбирается такой эталон S_v, для которого величина $F = |R(z,i) - R(v,i)| + |R(z,j) - R(v,j)|$ минимальна. Чтобы получить подтверждение того, что найденное решение является оптимальным, нужно продолжать описанные процедуры выбора очередного локатора и сокращения списка претендентов до тех пор, пока в этом списке не останется один или несколько самых близких образов, равноудаленных от точки Z.

Схематически этот алгоритм выглядит следующим образом.

Шаг 0. Вычисляется расстояние $R(z,i)$ до случайно выбранного образа S_i.

Шаг 1. По i-й строке матрицы M находятся образы, расстояния которых от образа S_i больше величины $2R(z,i)$, и эти образы исключаются из дальнейшего рассмотрения. В списке конкурентов образу S_i остается k_1 образов.

Шаг 2. Если $k_1 = 0$, то образ S_i объявляется ближайшим аналогом контрольного объекта Z.

Шаг 3. Если $k_1 = 1$, то оценивается расстояние $R(z,j)$ от Z до этого единственного конкурента S_j. Из двух образов S_i и S_j аналогом объекта Z считается образ, ближайший к Z.

Шаг 4. Если $k_1 > 1$, то по расстоянию $2R(z,j)$ список образов сокращается до величины $k_2 \le k_1$.

Шаг 5. Если $k_2 < 2$, то решение принимается по условиям шагов 3 или 4.

Шаг 6. Если $k_2 > 1$, то по матрице M выбирается такой образ S_v, при котором величина $F = |R(z,i) - R(v,i)| + |R(z,j) - R(v,j)|$ минимальна, и с его участием повторяются процедуры, аналогичные шагам 1–6: вычисление расстояния $R(z,v)$, сокращение списка конкурентов по критерию $2R$ и проверка числа остающихся конкурентов.

Шаг 7. Процесс останавливается, если принимается решение о принадлежности объекта Z одному из K известных образов.

В результате экспериментов выяснилось, что эффективность алгоритма почти не зависти от D, линейно зависит от K, но очень сильно зависит от размерности того пространства N, в котором измерялись парные расстояния. Это иллюстрируется таблицей ниже, полученной при $K = 1000$ и $D = 10$.

N	1	2	3	4	5	6	6	7	8	9	10
Q_k	3	4	6	10	20	37	76	120	210	360	890

В пространстве размерности $N > 20$ расстояния между всеми парами образов мало отличаются друг от друга, и по этой причине сфера с радиусом $2R$ охватывает эталоны практически всех образов. Из списка претендентов на каждом шаге вычеркивается только очередной локатор, и ускорения процесса поиска ближайшего аналога не происходит. Но в исходном пространстве малой размерности эффективность этого алгоритма очень высока. Здесь поиск при большом числе образов может ускоряться на 2–3 порядка.

5. Сокращение памяти

Недостатком трех последних алгоритмов является необходимость хранить в памяти большую матрицу парных расстояний ($K \times K/2$ чисел). Эксперименты показали, что в алгоритмах ЛОКАТОР-6 и ЛОКАТОР-7 при направленном выборе следующего локатора процесс сходится уже после использования не более 15 локаторов. Если их выбирать случайно, то количество требуемых локаторов увеличивается, но незначительно — до 20–25. На этом основании для указанных двух алгоритмов можно использовать не всю матрицу M парных расстояний, а $Q \ll K$ ее строк. При этом память сокращается в K/Q раз.

Можно предложить несколько вариантов стратегии выбора Q опорных локаторов. Представляется целесообразным выбирать такие эталоны-локаторы, распределение которых в пространстве X было бы согласовано с распределением всех эталонов. Это условие выполняется, если в списке образов номера V опорных векторов выбирать датчиком случайных чисел с равномерным распределением в диапазоне от 0 до K.

Что касается алгоритма ЛОКАТОР-8 $(r, -, 1)$, то количество необходимых локаторов сильно зависит от того, насколько велика дисперсия значений парных расстояний в матрице M. А эта дисперсия обратно пропорциональна размерности N пространства X, в котором измерялись эти парные расстояния: с ростом N количество локаторов Q растет неприемлемо быстро. По этой причине алгоритм ЛОКАТОР-8 будет эффективно работать только в пространствах Y, порожденных малоразмерными пространствами X (до $N < 10$).

Если распознавание ведется в изначально беспризнаковом пространстве по парным расстояниям, которые получены, например, методом экспертного оценивания, то ситуация выглядит вполне приемлемой. Дело в том, что эксперты не в состоянии делать количественные оценки мер различия между объектами, если число признаков больше нескольких единиц. В результате матрица M будет соответствовать ситуации с малыми значениями N, и алгоритм ЛОКАТОР-8 может оказаться вполне приемлемым для этого практически важного круга задач. Так, при $K = 1000$, $D = 10$, $N = 5$ при направленном поиске локаторов $Q = 20 \div 25$, а при случайном выборе локаторов $Q = 30 \div 40$. Следовательно, и для этого алгоритма можно хранить в памяти не всю матрицу $K \times K$, а полосу размером $Q \times K$, где $Q < 50$.

6. Оценки трудоемкости алгоритмов семейства ЛОКАТОР

Введем следующие обозначения:

c_1 — трудоемкость простых арифметических операций таких, как $+$, $-$, *abs*, *сравнение*;

c_2 — трудоемкость более сложных арифметических операций $*$, $/$, ...[2];

c_3 — трудоемкость операций работы со списком: *вычеркивание* и *добавление*;

$c_4 = \{c_1$ для метрики L_1, $c_1 + c_2$ для метрики $L_2\}$;

c_5 — затраты, связанные с оценкой расстояния от объекта Z до одного локатора;

L — трудоемкость для конкретного (среднестатистического) случая:

L^* — трудоемкость при самом плохом сценарии развития событий.

Введем некоторые вспомогательные величины:

$Q_N \le N$ — количество использованных признаков;

$Q_K \le K$ — количество использованных локаторов;

$Q_H \le H$ — количество шагов подбора порога;

k_t — число претендентов, оставшихся на t-м шаге ($k_0 = K$);

$K'_t = k_{t-1} - k_t$ — число претендентов, вычеркнутых на t-м шаге.

Трудоемкости разработанных алгоритмов в этих обозначениях оцениваются по следующим формулам.

ЛОКАТОР-1 $(L_\infty, d, -)$:

$$L_1 = \sum_{t=1}^{Q_N} k_{t-1}\big(3c_1 + K'_t c_3\big), \quad L_1^* = Q_N K 3c_1.$$

ЛОКАТОР-2 $(L_\infty, d, 1) + (L_\infty, -, 1)$:

$$L_2 = \sum_{i=1}^{Q_H} (L_1(d_i) + U(d_i)) \approx Q_H(L_1 + U),$$
$$U = \min\{U1, U2\}, \quad U_1 \le 5c_1 + 2c_2,$$
$$U_2 \le 2c_1 + c_2 + \sum_{d_i \le d_{\min}} (2c_1 + c_2) = (G_d + 1)(2c_1 + c_2),$$
$$L_2^* = Q_H\big(L_1^* + U\big).$$

ЛОКАТОР-3 $(L_2, d, -)$:

$$L_3 = \sum_{t=1}^{Q_N} k_{t-1}(2c_1 + c_4 + K'_t c_3), \quad L_3^* = Q_N K(2c_1 + c_4).$$

ЛОКАТОР-4 $(L_2, d, 1)$:

$$L_4 \le \sum_{t=1}^{Q_N} \big[k_{t-1}(4c_1 + c_4 + 2K'_t c_3\big) + 2(c_1 + c_2)\big],$$
$$L_4^* = Q_N[K(4c_1 + c_4) + 2(c_1 + c_2)].$$

ЛОКАТОР-5 $(L_2, -, 1)$:

$$L_5 = \sum_{t=1}^{Q_N} \big[2c_1 + c_2 + k_{t-1}\big(5c_1 + K'_t c_3\big)\big],$$
$$L_5^* = Q_N[2c_1 + c_2 + K(5c_1 + c_3)].$$

ЛОКАТОР-6 $(r, d, -)$:

$$L_6 = \sum_{t=1}^{Q_K} \big[F_t + 2(c_3 + c_1) + k_{t-1}\big(2c_1 + K'_t c_3\big)t\big],$$
$$F_t = k_{t-1}(3t+1)c_1 L_6^* = \sum\nolimits_{t=1}^{Q_K} \big[F_t^* + 2(c_3 + c_1) + 2c_1(K - t)\big],$$
$$F_t^* = (K - t)(3t + 1)c_1.$$

Определим долю F в полной трудоемкости L_6:

$$L_6^* \approx \sum_{t=1}^{Q_K} F_t^* + 2Q_K c_3 + Q_K(K - Q_K)c_1,$$
$$\sum_{t=1}^{Q_K} F_t^* \approx \frac{3}{2} Q_K^2 (2K - Q_K)c_1.$$

Из приведенных формул видно, что неслучайный выбор очередного локатора увеличивает количество операций C_1 в Q_K раз.

ЛОКАТОР-7 $(r, d, 1)$:

$$L_7 \le \sum_{t=1}^{Q_H} \left[Q_k\big(2c_1 + k_{t-1}\big(2c_1 + K'_t c_3\big)\big) + L_6(d_t) + U(d_t)\right] \approx Q_H\big(L'_6 + L_6 + U\big),$$

$$L^*_7 = (Q_H - 1)\big(K(K+1)2c_1 + L^*_6 + U\big) + L^*_6 + U.$$

ЛОКАТОР-8 $(r, -, 1)$:

$$L_8 = \sum_{t=1}^{Q_K}(F_t + 2c_3 + R_t), \quad R_t = \min\big\{R'_t, tR'_t\big\}, \quad R'_t = 2c_1 + k_t\big(2c_1 + K'_t c_3\big),$$

$$L^*_8 = \sum_{t=1}^{Q_K}\big(F^*_t + 2c_3 + R^*_t\big), \quad R^*_t = 2t(K - t + 1)c_1.$$

6. Выводы

Основной источник ускорения процесса поиска ближайшего аналога состоит в применении процедур поэтапного вычеркивания тех образов, которые на последующих шагах алгоритма не смогут претендовать на роль ближайшего аналога.

Гипотеза о том, что для ускорения поиска ближайшего аналога нужно переходить к использованию матрицы парных расстояний, не подтвердилась. Сравнение трудоемкости алгоритмов, работающих в пространствах X и Y, показало, что однотипные задачи (например, задачи $(L_\infty, d, 1)$ и $(r, d, 1)$ или $(L_2, -, 1)$ и $(r, -, 1)$) в пространстве X решаются в 3–10 раз быстрее, чем в пространстве Y, даже без учета разовых затрат на отображение X в Y. Отсюда следует, что, если образы и объекты описаны признаками пространства X, то переходить в пространство Y нецелесообразно. Алгоритмы ЛОКАТОР-6, -7, -8 нужно применять только в тех случаях, когда исходная информация задана сразу матрицей парных расстояний.

В пространстве Y применение направленного выбора очередного локатора по сравнению с его случайным выбором приводит к заметному замедлению работы алгоритмов. Однако при этом количество используемых локаторов уменьшается и, следовательно, сокращается количество процедур измерения расстояний до выбранных локаторов. При экспертном оценивании затраты на эти процедуры могут оказаться несопоставимо бóльшими по сравнению с затратами на направленный выбор локаторов. По этой причине в реализованных программах предусмотрены возможности пользоваться как направленным, так и случайным методом выбора очередного локатора. Можно ожидать, что направленный выбор будет использоваться чаще случайного.

В зависимости от параметров задачи выигрыш во времени, достигаемый разработанными алгоритмами семейства ЛОКАТОР, может составлять от нескольких раз до нескольких порядков.

ЛИТЕРАТУРА

1. *Загоруйко Н. Г., Дюбанов В. В.* Методы ускорения процесса поиска ближайшего аналога при распознавании большого числа образов // Автометрия. 2004. Т. 40, № 6. С. 101–109.

г. Новосибирск
Институт математики
им. С. Л. Соболева СО РАН
E-mail: zag@math.nsc.ru

Статья поступила 16 декабря 2005 г.
Окончательный вариант 7 августа 2006 г.