Об оптимальном нелинейном растяжении матриц самосравнения одномерных сигналов

А. А. Петюшко, И. Л. Мазуренко
(МГУ имени М. В. Ломоносова, Москва)

В работе рассматриваются матрицы самосравнения одномерного сигнала (в частности, речевого). Предлагается метод нелинейного растяжения этих симметричных матриц для нахождения оптимального расстояния между ними в смысле похожести сигналов.

Ключевые слова: одномерный сигнал, матрица самосравнения, нелинейное растяжение.

В данной работе рассмотрен вопрос нахождения оптимального растяжения матриц самосравнения одномерных сигналов (например, речевых) для подсчёта меры близости между ними с целью повысить надёжность результатов распознавания. Заметим, что кратко этот метод был упомянут в статье [1], а также описан в составе программного комплекса для повышения надёжности распознавания речи в статье [2].

Традиционный подход решения задачи автоматического распознавания речевых команд заключается в сравнении речевого сигнала с эталонами команд, фонем и т. п. Такое сравнение делается с помощью метода динамического программирования (см. [3]). Распространённым походом к реализации автоматического распознавания речи является метод динамической деформации времени [4].

Для увеличения надёжности распознавания предлагается применить метод динамического программирования для нахождения расстояния не между одномерными сигналами, а между матрицами их самосравнения.

Дадим конструктивное определение **матрицы самосравнения** S для одномерного сигнала $s \in \mathbb{R}^N$:

1) Сигнал разбивается на n окон одинаковой длины (возможно, частично перекрывающихся).

2) В каждом окне с номером $i = 1 \dots n$ вычисляется вектор признаков $v_i(s) \in \mathbb{R}^k$ (спектральных, кепстральных, коэффициентов линейного предсказания и т. п. — набор признаков выбирается в зависимости от конкретной реализации).
3) На основе некоторой меры близости $\rho_k(.,.)$ (например, евклидовой метрики в пространстве $\mathbb{R}^k$) вычисляется расстояние для каждой пары векторов из соответствующих окон.
4) На место (i, j) в матрице самосравнения заносится вычисленное в п. 3 расстояние между i-м и j-м вектором признаков: $S(i, j) = \rho_k(v_i(s), v_j(s))$.

Таким образом, для сигнала s, которому соответствуют n векторов признаков $v_i(s)$ для $i = 1 \dots n$, получим матрицу самосравнения S размера $n \times n$.

По аналогии с методом динамической деформации времени для одномерного сигнала, авторами предлагается использовать метод нелинейного сжатия-растяжения матриц. Необходимо таким образом увеличить две сравниваемые матрицы (будем именно *растягивать* матрицы) до одинакового размера, чтобы, с одной стороны, это растяжение минимизировало расстояние между этими матрицами, а с другой стороны, после растяжения можно было бы использовать некоторую функцию расстояния между матрицами одинаковой размерности.

Опишем процедуру для растяжения двух матриц самосравнения S_1, S_2 (размера $n_1 \times n_1$ и $n_2 \times n_2$ соответственно) для нахождения расстояния между ними более строго. Для этого строится квадратная таблица — **таблица растяжений** (или **таблица пути**) R размера $n_1 \times n_2$, элементом (i, j) которой является пара: 1) последовательность растяжений $Rp(i, j)$, минимизирующая расстояние между квадратными подматрицами, левый верхний угол которых совпадает с верхним левым углом для исходных матриц S_1, S_2, размера $i \times i$ для S_1 и $j \times j$ для S_2 соответственно, и 2) само это расстояние $Rd(i, j)$.

Отметим, что расстояние $Rd(i, j)$ между квадратными подматрицами размера $i \times i$ для S_1 и $j \times j$ для S_2 — это действительное число, то есть $Rd(i, j) \in \mathbb{R}$. Последовательность же растяжений $Rp(i, j)$ — это упорядоченный набор пар $Rp(i, j) = ((a_1, b_1), \dots, (a_m, b_m))$ таких, что $(a_1, b_1) = (1, 1)$, $(a_m, b_m) = (i, j)$, и $(a_k, b_k) \neq (a_{k+1}, b_{k+1})$, причём $0 \leqslant a_{k+1} - a_k \leqslant 1$ и $0 \leqslant b_{k+1} - b_k \leqslant 1$ для любого $k = 1, \dots, m - 1$.

Предположим, что мы используем меру близости $f_k(.,.)$ для нахождения расстояния между двумя векторами размерности k (заметим, что в общем случае $f_k(.,.)$ и $\rho_k(.,.)$ — разные функции). Также будем использовать обозначения $e_1, e_2, \ldots$ для обозначения базисных векторов в многомерном пространстве. Тогда первая строка и первый столбец таблицы растяжений R формируются следующим образом:

$$Rd(1,1) = f_1(S_1(1,1), S_2(1,1)), Rp(1,1) = (1,1),$$
$$Rd(1,k) = f_k\left(S_1(1,1)\sum_{i=1}^{k} e_i,\ \sum_{i=1}^{k} S_2(1,i)e_i\right) + Rd(1,k-1),\ k > 1,$$
$$Rp(1,k) = ((1,1),\ldots,(1,k)),\ k = 2,\ldots,n_2$$
$$Rd(m,1) = f_m\left(\sum_{i=1}^{m} S_1(1,i)e_i,\ S_2(1,1)\sum_{i=1}^{m} e_i\right) + Rd(m-1,1),\ m > 2,$$
$$Rp(m,1) = ((1,1),\ldots,(m,1)),\ m = 2,\ldots,n_1.$$

Содержательно это значит, что левый верхний элемент матрицы S_1 копируется на все координаты вектора длины k, и на место $(1,k)$ таблицы R в качестве Rd заносится сумма расстояния между этим вектором и вектором такой же длины, являющемся началом k-ой строки второй матрицы S_2 длины k, и числа $Rd(1,k-1)$ из клетки слева (для первого столбца все аналогично с точностью до замены S_1 на S_2). Одновременно в каждую клетку R в качестве Rp заносится путь, по которому мы в неё пришли, то есть последовательность таких пар чисел, в которых первое число отвечает за номер строки (или столбца, что то же самое, так как матрицы самосравнения симметричны) первой матрицы S_1, а второе — за номер строки (столбца) второй матрицы S_2.

Для остальных клеток таблицы растяжений используем следующую итеративную процедуру:

1) Пусть надо посчитать путь и расстояние для клетки с номером (i,j). Предположим, что для всех других клеток с индексами (i_1,j_1) таких, что $(i_1,j_1) \neq (i,j)$, $1 \leqslant i_1 \leqslant i, 1 \leqslant j_1 \leqslant j$, мы знаем $Rp(i_1,j_1), Rd(i_1,j_1)$ (как заполнить первую строку и первый столбец, было показано выше).
2) Рассмотрим три соседние уже заполненные клетки: с индексами $(i-1,j-1)$, $(i,j-1)$ и $(i-1,j)$. Рассчитаем три значения $d_{1,1}$, $d_{0,1}$ и $d_{1,0}$, которые будут соответствовать расстоянию между квадратными подматрицами размера $i \times i$ для S_1 и $j \times j$ для S_2 при разных последовательностях растяжений этих подматриц, а именно:

- $d_{1,1}$: одновременное растяжение на 1 клетку обеих подматриц S_1 и S_2,
- $d_{0,1}$: растяжение на 1 клетку только подматрицы S_2,
- $d_{1,0}$: растяжение на 1 клетку только подматрицы S_1.

3) Для примера рассчитаем значение $d_{1,1}$ ($d_{0,1}$ и $d_{1,0}$ рассчитываются аналогично). Пусть $Rp(i-1, j-1) = (a_1, b_1), \ldots, (a_m, b_m)$. Тогда, если обозначить через

$$v_1 = S_1(i,i)e_{m+1} + \sum_{k=1}^{m} S_1(i, a_k)e_k,$$
$$v_2 = S_2(j,j)e_{m+1} + \sum_{k=1}^{m} S_2(j, b_k)e_k,$$

то расстояние между квадратными подматрицами размера $i \times i$ для S_1 и $j \times j$ для S_2 при одновременном растяжении на 1 клетку обеих подматриц будет равно

$$d_{1,1} = Rd(i-1, j-1) + f_{m+1}(v_1, v_2).$$

Заметим, что с помощью такой процедуры расчёта значений расстояний между квадратными подматрицами разного размера мы учитываем все прошлые растяжения, которые были записаны в предыдущую клетку пути, и альтернатива возникает только в выборе растяжения последней клетки (растягиваем обе подматрицы сразу на 1 клетку, или только одну подматрицу; предыдущая последовательность растяжений сохраняется).

4) Пусть $\min(d_{1,1}, d_{0,1}, d_{1,0}) = d_{a,b}$, где $0 \leqslant a, b \leqslant 1$. Тогда

$$Rd(i,j) = d_{a,b}, Rp(i,j) = (Rp(i-a, j-b), (i,j)).$$

В итоге в последней (правой нижней) ячейке таблицы растяжений мы получим путь $Rp(n_1, n_2)$ — последовательность растяжений, которая минимизирует расстояние между матрицами самосравнения S_1, S_2.

Как следствие из вышеописанной процедуры нахождения оптимального растяжения симметричных матриц имеем следующую теорему.

Теорема 1. *Сложность алгоритма (в терминах количества операций взятия минимума и нахождения расстояния между векторами) нахождения оптимального растяжения двух симметричных матриц самосравнения S_1, S_2 размеров $n_1 \times n_1$ и $n_2 \times n_2$ соответственно равна $O(n_1 n_2)$.*

Опишем процедуру построения растянутых матриц T_1, T_2, которые получаются применением последовательности растяжений $Rp(n_1, n_2)$ к изначальным матрицам самосравнения S_1, S_2. Пусть $Rp(n_1, n_2)$ содержит m пар индексов: $Rp(n_1, n_2) = ((a_1, b_1), \ldots, (a_m, b_m))$. Тогда матрицы T_1, T_2 имеют одинаковый размер $m \times m$ и их элементы определяются следующим образом ($1 \leqslant i, j \leqslant m$):

$$T_1(i, j) = S_1(a_i, a_j), T_2(i, j) = S_2(b_i, b_j).$$

Напоследок заметим, что в качестве расстояния $r(S_1, S_2)$ между симметричными матрицами самосравнения S_1, S_2 предлагается брать не значение $Rd(n_1, n_2)$, а некоторую матричную норму $g(.)$ от разности растянутых матриц T_1, T_2 одного размера:

$$r(S_1, S_2) = g(T_1 - T_2).$$

Такой матричной нормой может служить, например, матричная норма Фробениуса.

Список литературы

[1] Бабин Д. Н., Дементиенко В. В., Мазуренко И. Л., Миргородский В. И., Михайлов Д. С., Холоденко А. Б., Уранцев А. В. Система речевого контроля состояния машиниста // Интеллектуальные системы. Теория и приложения. — 2014. — Т. 18. № 4. — С. 69–75.

[2] Мазуренко И. Л., Петюшко А. А. Метод оптимального нелинейного растяжения симметричных матриц в задачах распознавания // Интеллектуальные системы. Теория и приложения. — В печати.

[3] Bellman R. On the theory of dynamic programming // Proceedings of the National Academy of Sciences. — 1952. — Vol. 38. № 8. — P. 716–719.

[4] Винцюк Т. К. Распознавание слов устной речи методами динамического программирования // Кибернетика. — 1968. — № 1. — С. 81–88.