ИНФОРМАТИКА

2010 июль-сентябрь № 3

УДК 517.514

В.С. Муха

ЛИНЕЙНОЕ ПРОКРУСТОВО ПРЕОБРАЗОВАНИЕ ДВУХМЕРНЫХ МАТРИЦ

Формулируется и решается задача линейного двухмерно-матричного прокрустова преобразования. Разрабатывается программная реализация алгоритма прокрустова преобразованиия. Выполняется сравнение результатов работы программы с существующей в системе программирования Matlab программой procrustes.

Введение

Согласно [1] термин «прокрустов анализ» впервые был введен J.R. Hurley и R.B. Cattell [2] для методов сравнения различных множеств главных компонент. Такое название методы получили по имени известного в греческой мифологии трактирщика Прокруста. Прокруст укладывал путников в ложе и ночью во время сна подгонял их размеры под размер ложа, обрубая слишком высоких путников и вытягивая слишком низких. Конец разбою Прокруста положил Тесей, уложивший Прокруста в его собственное ложе и отрубивший ему выступающие голову и ноги.

В настоящее время прокрустово преобразование (Procrustes transformation) находит широкое применение для сравнения различных родственных множеств данных во многих предметных областях [1]. Одно из таких применений – идентификация лиц в криминалистике [1].

Классической считается ортогональная прокрустова задача, впервые сформулированная и решенная в [3]. Имеются также различные ее обобщения [4–6].

В доступной по данной проблеме литературе просматриваются идея прокрустова преобразования и некоторые конечные результаты, однако для решения конкретных задач этого недостаточно. Возникает потребность в модификациях и обобщениях данного подхода, которые невозможны без знания теоретических основ. С целью восполнения пробелов, существующих в данной области в отечественной литературе, в статье сформулирована и решена задача линейного двухмерно-матричного прокрустова преобразования.

Постановка задачи

Идея прокрустова преобразования состоит в том, чтобы линейным преобразованием (сдвигом, вращением, масштабированием) наилучшим образом подогнать матрицу данных x к матрице данных y.

Ортогональная прокрустова задача [5, 6] состоит в преобразовании известной матрицы x с помощью ортогональной матрицы преобразования T с целью минимизировать сумму квадратов остаточной матрицы $v = xT - y$:

$$tr(v'v) \to \min_T ,$$

где y – другая заданная матрица, при условии ортогональности матрицы T:

$$T'T = TT' = I .$$

Для ее решения необходимо минимизировать функцию Лагранжа

$$F = tr(v'v) + tr(L(T'T - I)),$$

где L – матрица множителей Лагранжа.

Ортогональная прокрустова задача имеет следующие особенности: в постановке задачи отсутствует преобразование-сдвиг; матрица преобразования T предполагается ортогональной. Отсутствие сдвига не позволяет обоснованно его выбирать в задачах, где он присутствует. Требование ортогональности является достаточно специфическим для его применения в общем случае. В данной статье снимаются указанные ограничения, т. е. рассматривается линейное преобразование общего вида без предположения ортогональности.

Итак, предположим, что заданы две матрицы данных: $x=(x_{i,\mu})$ и $y=(y_{i,\mu})$, $i=\overline{1,k}$, $\mu=\overline{1,n}$. Преобразуем матрицу x в матрицу $d=(d_{i,\mu})$ с помощью линейного преобразования

$$d=c+c_{(1,1)}x\,, \tag{1}$$

обозначим

$$z=y-d=y-c-c_{(1,1)}x$$

и поставим задачу отыскания таких коэффициентов c, $c_{(1,1)}$ преобразования (1), при которых величина $r=\|z\|_E^2=\sum_{i=1}^{k}\sum_{\mu=1}^{n}z_{i,\mu}^2=\|y-d\|_E^2$ будет минимальной:

$$r=\|z\|_E^2=\sum_{i=1}^{k}\sum_{\mu=1}^{n}z_{i,\mu}^2\to\min_{c,c_{(1,1)}}. \tag{2}$$

Назовем преобразование (1) с полученными оптимальными коэффициентами двухмерно-матричным линейным прокрустовым преобразованием, а минимальное значение $r_{\min}$ критерия (2) – прокрустовым расстоянием между x и y.

В такой постановке задача является тривиальной. Действительно, выбрав $c=y$, $c_{(1,1)}=0$, получим $r_{\min}=0$. Очевидно, что в задачу необходимо вводить некоторые дополнительные ограничения. Поскольку преобразованию подлежит матрица x в целом, ясно, что сдвиг должен быть одинаковым для всех вектор-столбцов $x_\mu=\left(x_i\right)_\mu=\left(x_{i,\mu}\right)$ матрицы x, т. е. матрица сдвига c должна состоять из одинаковых столбцов. С учетом данного ограничения задача может быть сформулирована и решена в следующей интерпретации.

Будем рассматривать отдельные столбцы $x_\mu=\left(x_i\right)_\mu=\left(x_{i,\mu}\right)$, $y_\mu=\left(y_i\right)_\mu=\left(y_{i,\mu}\right)$, $d_\mu=\left(d_i\right)_\mu=\left(d_{i,\mu}\right)$ матриц x, y, d. Тогда преобразование (1) можно представить в виде

$$d_\mu=c_{(1,0)}+c_{(1,1)}x_\mu\,,\ \mu=\overline{1,n}\,, \tag{3}$$

где $c_{(1,0)}$ – вектор-столбец сдвига, а критерий оптимальности (2) – в виде

$$r=\|z\|_E^2=\sum_{i=1}^{k}\sum_{\mu=1}^{n}z_{i,\mu}^2\,.$$

Поставим задачу отыскания вектор-столбца $c_{(1,0)}$ и матрицы $c_{(1,1)}$ модели данных (3), доставляющих минимум критерию (2), т. е. оптимизационную задачу вида

$$r=\|z\|_E^2=\sum_{i=1}^{k}\sum_{\mu=1}^{n}z_{i,\mu}^2=\sum_{i=1}^{k}\sum_{\mu=1}^{n}(y_{i,\mu}-d_{i,\mu})^2\to\min_{c_{(1,0)},c_{(1,1)}}. \tag{4}$$

В данной постановке задачи учтено указанное выше ограничение на матрицу c преобразования (1), ибо после объединения выражений (3) в единую матричную запись (1) матрица c окажется состоящей из одинаковых столбцов вида $c_{(1,0)}$.

Решение задачи

Для решения задачи целесообразно применять многомерно-матричный подход [7]. В связи с этим запишем модель данных (3) в многомерно-матричной форме:

$$d_\mu = c_{(1,0)} + {}^{0,1}\left(c_{(1,1)} x_\mu\right),\ \mu = \overline{1,n}, \tag{5}$$

где левые верхние индексы $0,1$ означают $(0,1)$-свернутое произведение [7]. С учетом принятых обозначений легко обнаружить, что оптимизационная задача (5), (4) является одномерно-матричным линейным случаем многомерно-матричной полиномиальной регрессионной задачи, рассмотренной в работе [8]. Полагая в линейном решении работы [8] $p = q = 1$, получим для оптимальных параметров $c_{(1,0)}$, $c_{(1,1)}$ модели данных (5) следующие выражения:

$$\widehat{c}_{(1,1)} = {}^{0,1}(\overset{\circ}{s}_{yx}\ {}^{0,1}(\overset{\circ}{s}_{x^2})^{-1}; \tag{6}$$

$$\widehat{c}_{(1,0)} = \overline{y} - {}^{0,1}(\,{}^{0,1}(\overset{\circ}{s}_{yx}\ {}^{0,1}(\overset{\circ}{s}_{x^2})^{-1})\overline{x}), \tag{7}$$

где

$$\overline{x}_c = \frac{1}{n}\sum_{\mu=1}^{n} x_\mu,\ \overline{y}_c = \frac{1}{n}\sum_{\mu=1}^{n} y_\mu; \tag{8}$$

$$\overset{\circ}{s}_{x^2} = \frac{1}{n}\sum_{\mu=1}^{n} {}^{0,0}\left((x_\mu - \overline{x}_c)(x_\mu - \overline{x}_c)\right),\ \overset{\circ}{s}_{yx} = \frac{1}{n}\sum_{\mu=1}^{n} {}^{0,0}\left((y_\mu - \overline{y}_c)(x_\mu - \overline{x}_c)\right).$$

Предполагается, что матрица $\overset{\circ}{s}_{x^2}$ не вырожденная, т. е. $(0,1)$-обратная матрица ${}^{0,1}(\overset{\circ}{s}_{x^2})^{-1}$ существует. Необходимым для этого является условие $k < n$. В случае вырожденности матрицы $\overset{\circ}{s}_{x^2}$ рекомендуется применять псевдообращение.

Подставляя (6), (7) в (5), получим

$$d_\mu = \overline{y}_c + {}^{0,1}\left({}^{0,1}(\overset{\circ}{s}_{yx}\ {}^{0,1}(\overset{\circ}{s}_{x^2})^{-1})(x_\mu - \overline{x}_c)\right),\ \mu = \overline{1,n}. \tag{9}$$

Объединяя выражения (9) в единую матричную запись вида (1), получим следующее выражение линейного прокрустова преобразования:

$$d = \overline{y} + {}^{0,1}\left({}^{0,1}(\overset{\circ}{s}_{yx}\ {}^{0,1}(\overset{\circ}{s}_{x^2})^{-1})(x - \overline{x})\right), \tag{10}$$

где $\overline{y}$ и $\overline{x}$ – матрицы, состоящие из одинаковых столбцов $\overline{y}_c$ и $\overline{x}_c$ соответственно: $\overline{y} = (\overline{y}_c, \overline{y}_c, ..., \overline{y}_c)$, $\overline{x} = (\overline{x}_c, \overline{x}_c, ..., \overline{x}_c)$.

Отметим, что выражения для $\overset{\circ}{s}_{x^2}$ и $\overset{\circ}{s}_{yx}$ в прокрустовом преобразовании (10) допускают следующее представление:

$$\overset{\circ}{s}_{x^2} = \frac{1}{n}\sum_{\mu=1}^{n}{}^{0,0}\left((x_\mu - \overline{x}_c)(x_\mu - \overline{x}_c)\right) = \frac{1}{n}{}^{0,1}\left((x-\overline{x})(x-\overline{x})^T\right) = \frac{1}{n}{}^{0,1}(\overset{\circ}{x}(\overset{\circ}{x})^T)\,; \qquad (11)$$

$$\overset{\circ}{s}_{yx} = \frac{1}{n}\sum_{\mu=1}^{n}{}^{0,0}\left((y_\mu - \overline{y}_c)(x_\mu - \overline{x}_c)\right) = \frac{1}{n}{}^{0,1}\left((y-\overline{y})(x-\overline{x})^T\right) = \frac{1}{n}{}^{0,1}(\overset{\circ}{y}(\overset{\circ}{x})^T)\,, \qquad (12)$$

где $\overset{\circ}{x} = x - \overline{x}$, $\overset{\circ}{y} = y - \overline{y}$ и T – подстановка транспонирования двухмерной матрицы. Отметим также, что все матрицы в выражениях (10)–(12) двухмерные, так что $(0,1)$-свернутые произведения являются известными произведениями векторно-матричного подхода. Учитывая представления (11), (12) и опуская в (10)–(12) символ $(0,1)$-свернутого умножения, получим линейное двухмерно-матричное прокрустово преобразование в известных векторно-матричных обозначениях:

$$d = \overline{y} + \overset{\circ}{y}(\overset{\circ}{x})^T(\overset{\circ}{x}(\overset{\circ}{x})^T)^{-1}\overset{\circ}{x}\,. \qquad (13)$$

Прокрустово расстояние $r_{\min}$ может быть найдено непосредственно по формуле (2) после расчета матрицы d (10).

Таким образом, доказана следующая теорема.

Теорема*. Если $x = (x_{i,\mu})$ и $y = (y_{i,\mu})$, $i = \overline{1,k}$, $\mu = \overline{1,n}$, – две $(k \times n)$-матрицы данных, $k < n$, то линейное преобразование $d = c + c_{(1,1)}x$ с матрицей c, состоящей из одинаковых столбцов, и произвольной матрицей $c_{(1,1)}$, обеспечивающее минимум критерия $r = \|y - d\|_E^2$, определяется матричным выражением (13), в котором $\overset{\circ}{x} = x - \overline{x}$, $\overset{\circ}{y} = y - \overline{y}$, $\overline{x} = (\overline{x}_c, \overline{x}_c, ..., \overline{x}_c)$, $\overline{y} = (\overline{y}_c, \overline{y}_c, ..., \overline{y}_c)$, $\overline{x}_c = (\overline{x}_{c,i}) = \frac{1}{n}\sum_{\mu=1}^{n} x_{i,\mu}$, $\overline{y}_c = (\overline{y}_{c,i}) = \frac{1}{n}\sum_{\mu=1}^{n} y_{i,\mu}$.*

Данная теорема определяет следующий алгоритм прокрустова преобразования:

1) рассчитываются векторы $\overline{x}_c$, $\overline{y}_c$ по формулам (8);

2) формируются матрицы $\overline{x} = (\overline{x}_c, \overline{x}_c, ..., \overline{x}_c)$, $\overline{y} = (\overline{y}_c, \overline{y}_c, ..., \overline{y}_c)$;

3) рассчитываются матрицы $\overset{\circ}{x} = x - \overline{x}$, $\overset{\circ}{y} = y - \overline{y}$;

4) рассчитывается матрица d (13);

5) рассчитывается прокрустово расстояние $r_{\min} = \|y - d\|_E^2$.

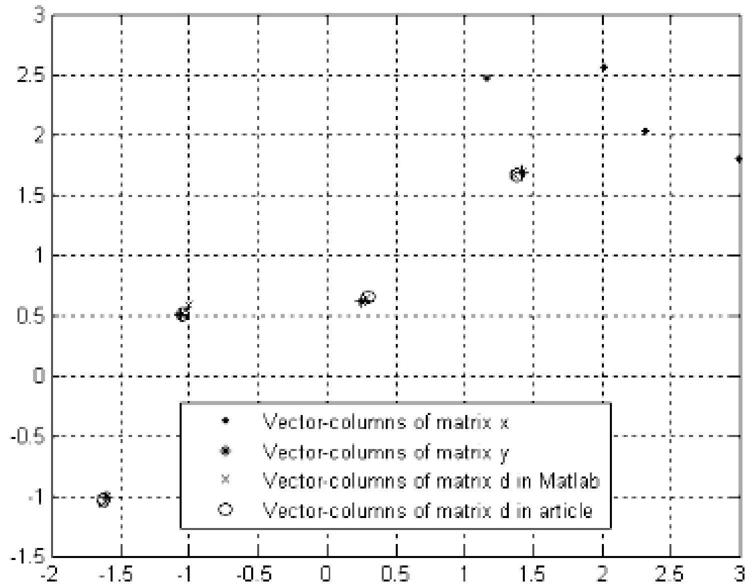
Пример

В системе программирования Matlab имеется программа-функция procrustes, обеспечивающая прокрустово преобразование со сдвигом, вращением и масштабированием. Поэтому целесообразно сравнить работу представленного в статье алгоритма с работой функции procrustes. С этой целью была написана *m*-файл-функция, реализующая алгоритм линейного прокрустова преобразования. Результаты работы этой функции и функции procrustes на исходных данных

$$x = \begin{pmatrix} 1{,}1614 & 2{,}3145 & 2{,}0105 & 2{,}9929 \\ 2{,}4704 & 2{,}0294 & 2{,}5655 & 1{,}7955 \end{pmatrix}, \quad y = \begin{pmatrix} -1{,}6041 & 0{,}2573 & -1{,}0565 & 1{,}4151 \\ -1{,}0106 & 0{,}6145 & 0{,}5077 & 1{,}6924 \end{pmatrix}$$

представлены на рисунке. Визуально результаты работы обеих функций весьма близки. Вместе с тем предложенный алгоритм обеспечил меньшее прокрустово расстояние $r_{\min} = 0{,}0062$ между

x и y по сравнению с $r_{\min} = 0,0189$ для функции procrustes. Это дополнительно подтверждает правильность и достоверность полученных в данной статье результатов.



Результаты прокрустова преобразования предложенным алгоритмом и программой procrustes

В примере к описанию программы procrustes матрицы x и y размером (2×10) формируются из случайных чисел, так что различные прогоны примера демонстрируют работу программы на различных исходных данных. Многочисленные прогоны этого примера неизменно показывают меньшее прокрустово расстояние у предложенного алгоритма по сравнению с алгоритмом программы procrustes. На некоторых матрицах x и y разница в расстояниях оказывается достаточно ощутимой.

Заключение

В статье дана строгая математическая постановка задачи линейного двухмерно-матричного прокрустова преобразования и метод ее решения, получен и реализован программно алгоритм прокрустова преобразования, выполнено численное сравнение работы алгоритма с известным алгоритмом системы программирования Matlab, подтвердившее работоспособность полученного алгоритма. Достоинством алгоритма является возможность его реализации в любой системе программирования. Алгоритм может быть использован в задачах распознавания образов.

Список литературы

1. Procrustes rotation in analytical chemistry, a tutorial / J.M. Andrade [et al.] // Chemometrics and Intelligent Laboratory Systems. – 2004. – № 72. – P. 123–132.

2. Hurley, J.R., Cattell, R.B. // Behav. Sci. – 1962. – № 7. – P. 258–262.

3. Schonemann, P.H. A generalized solution of the orthogonal procrustes problem / P.H. Schonemann // Psychometrica. – 1966. – Vol. 31. – № 1. – P. 1–10.

4. Gower, J.C. Generalized Procrustes analysis / J.C. Gower // Psychometrika. – 1975. – Vol. 40. – P. 33–51.

5. Crosilla, F. Procrustes Analysis and Geodetic Sciences. Technical report. Part 1 / F. Crosilla. – Stuttgart, Germamy : Univ. of Stuttgart, Dept. of Geodesy and Geoinformatics, 1999. – P. 69–78.

6. Generalized Procrustes analysis and its applications in photogrammetry // Presented to prof. Armin W. Gruen [Electronic resource]. – June, 2003. – Mode of access : http://www.photogrammetry.ethz.ch/general/persons/devrim/2003CH_Praktikum_Procrustes.pdf. – Date of access : 07.06.2010.

7. Муха, В.С. Анализ многомерных данных / В.С. Муха. – Минск : Технопринт, 2004. – 368 с.
8. Муха, В.С. Многомерно-матричный полиномиальный регрессионный анализ. Оценки параметров / В.С. Муха // Известия НАН Беларуси. Сер. физ.-мат. наук. – 2007. – № 1. – С. 45–51.

Поступила 17.02.10

Белорусский государственный университет
информатики и радиоэлектроники,
Минск, П. Бровки, 6
e-mail: mukha@.bsuir.by

V.S. Mukha

LINEAR PROCRUSTES TRANSFORMATION OF TWO-DIMENSIONAL MATRICES

The problem of linear Procrustes transformation of two-dimensional matrices is formulated and solved. The transformation is implemented in a software module. The results are compared with those obtained with the help of existing Matlab-based implementation.