МОДЕЛИРОВАНИЕ ДАННЫХ ПРИ ПОМОЩИ КРИВЫХ ДЛЯ ВОССТАНОВЛЕНИЯ ПРОБЕЛОВ В ТАБЛИЦАХ

Часть 1.

Пусть задана прямоугольная таблица $A = a_{ij}$. Каждая строка таблицы A представляет собой вектор данных. В каждом векторе могут присутствовать пропуски, которые обозначим @.

Необходимо решить следующие задачи:

1. Наиболее точно определить значения пропущенных данных.
2. При необходимости так изменить данные в таблице, чтобы они наилучшим образом соответствовали моделям, используемым для восстановления пропусков в таблице.
3. Построить алгоритм для обработки новых строк с пропусками, при условии что новые данные соответствуют тем же зависимостям, что и в таблице.

Для решения этих задач предлагается использовать метод последовательного приближения множества векторов данных прямыми вида $x_i y_j + b_j$, то есть необходимо найти наилучшее отображение таблицы A матрицей $P = x_i y_j + b_j$. Для этого используем метод наименьших квадратов.

Процедура нахождения матрицы приближения.

1. Необходимо минимизировать функционал

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - x_i y_j - b_j)^2 \to \min. \qquad (1)$$

2. При фиксированных векторах y_j и b_j значения x_i, доставляющие минимум форме (1), определяются из равенств $\partial\Phi/\partial x_i = 0$ следующим образом:

$$x_i = \left(\sum_{\substack{j \\ a_{ij} \neq @}} (a_{ij} - b_j) y_j \right) \Bigg/ \left(\sum_{\substack{j \\ a_{ij} \neq @}} (y_j)^2 \right). \qquad (2)$$

Начальные значения:

y – нормированное на 1 случайное значение, т.е. $\sum_j y_j^2 = 1$; (3)

b_j – среднее значение в столбце, т. е. $b_j = \frac{1}{n_j} \sum_{\substack{i \\ a_{ij} \neq @}} a_{ij}$, где $n_j = \sum_{\substack{i \\ a_{ij} \neq @}} 1$. (4)

3. По фиксированному x_i, можно сразу по явным формулам посчитать значения y_j и b_j, доставляющие минимум форме (1). Они определяются из двух равенств $\partial\Phi/\partial y_j = 0$ и $\partial\Phi/\partial b_j = 0$ следующим образом:

Для каждого j имеем систему из двух уравнений относительно y_j и b_j:

$$\begin{cases} y_j A_{01}^j + b_j A_{00}^j = B_o^j \\ y_j A_{11}^j + b_j A_{10}^j = B_1^j \end{cases}, \text{ где } A_{kl}^j = \sum_{\substack{i \\ a_{ij} \neq @}} x_i^{k+l},\ B_k^j = \sum_{\substack{i \\ a_{ij} \neq @}} a_{ij} x_i^k,\ k=0..1,\ l=0..1.$$

Выражая из первого уравнения b_j и подставляя полученное значение во второе, получим:

$$y_j = \frac{B_1^j - B_0^j \dfrac{A_{10}^j}{A_{00}^j}}{A_{11}^j - A_{01}^j \dfrac{A_{10}^j}{A_{00}^j}},\ b_j = \frac{B_0^j - y_j A_{01}^j}{A_{00}^j}. \qquad (5)$$

4. Критерий остановки – $\Delta\Phi/\Phi < \varepsilon$ или $\Phi < \delta$ для некоторых $\varepsilon, \delta < 0$, где $\Delta\Phi$ – полученное за итерацию уменьшение значения Ф (текущего значения).

Алгоритм решения задачи.

1. Пусть t - номер текущей итерации. A_0 - начальная матрица. По (3) и (4) инициализируем начальные значения y^0 и b^0.
2. По (2) и (5) находим значения векторов x^t, y^t и b^t.
3. Рассчитываем матрицу $P_t = x_i^t y_j^t + b_j^t$ - наилучшее приближение для текущей матрицы A_t.
4. По (1) вычисляем значение Φ_t и $\Delta\Phi = \Phi_t - \Phi_{t-1}$.
5. Если $\Phi_t < \delta$ или $\Delta\Phi/\Phi_t < \varepsilon$, то останов.
6. Иначе вычисляем $A_{t+1} = A_t - P_t$ и переход на шаг 2, при этом y^t, b^t и A_{t+1} - начальные значения для следующей итерации.

В результате работы данного алгоритма происходит последовательное исчерпание матрицы A. После q итераций будет построена последовательность матриц $P_t = x_i^t y_j^t + b_j^t,\ t = 1..q$ или $P = \sum_{t=1}^{q} P_t$, исчерпывающая исходную матрицу A с заданной точностью.

Таким образом, матрица P является решением поставленных задач.

1. Значения пропущенных данных определяются из значений соответствующих элементов матрицы P.
2. Для исправления исходной таблицы ее заменяют матрицей P.
3. Опишем операцию восстановления данных в поступающей на обработку строке a_j с пробелами (некоторые $(a_j = @)$). Для каждой матрицы P_t по заданной строке определим число $x^t(a)$ и вектор a_j^q:

$$a_j^0 = a_j \ \ (a_j \neq @);$$

$$x^1(a)=\left(\sum_{\substack{j\\ a_j\neq @}}(a_j^0-b_j^1)y_j^1\right)\Bigg/\left(\sum_{\substack{j\\ a_j\neq @}}(y_j^1)^2\right);$$

$$a_j^1=a_j^0-b_j^1-x^1(a)y_j^1\quad (a_j\neq @);$$

$$\ldots\ldots\ldots\ldots\ldots..$$

$$x^q(a)=\left(\sum_{\substack{j\\ a_j\neq @}}(a_j^{q-1}-b_j^q)y_j^q\right)\Bigg/\left(\sum_{\substack{j\\ a_j\neq @}}(y_j^q)^2\right);$$

$$a_j^q=a_j^{q-1}-b_j^q-x^q(a)y_j^q\quad (a_j\neq @);$$

Тогда для q-факторного восстановления данных:

$$a_j=\sum_{t=1}^{q}x^t(a)y_j^t+b_j^t,\ (a_j\neq @).$$

Если пробелы отсутствуют, то описанный метод приводит к обычным главным компонентам исходной таблицы данных. В этом случае, начиная с $t=2$, $P_t=x_i^t y_j^t$ ($b=0$). В общем случае это не так и центрирование к данным с пробелами неприменимо.

Также следует учесть, что при отсутствии пробелов, полученные прямые будут ортогональны, то есть получим ортогональную систему факторов (прямых). Исходя из этого, при неполных данных возможен процесс ортогонализации полученной системы факторов, который заключается в том, что исходная таблица восстанавливается при помощи полученной системы, после чего эта система пересчитывается заново, но уже на полных данных.

(Материал заимствован из статьи А.А.Россиева.)