АНАЛИЗ ДАННЫХ В ЭКСПЕРТНЫХ СИСТЕМАХ
(Вычислительные системы)
1986 год Выпуск 117

УДК 519.712.2

АЛГОРИТМ ВЫБОРА ЛИНЕЙНОЙ РЕГРЕССИИ МИНИМАЛЬНОЙ СЛОЖНОСТИ

В.Г.Устюжанинов, Е.Н.Шемякина

§1. Постановка задачи

Ряд прикладных задач [2,с. 315; 3; 6, с.158] могут быть сформулированы следующим образом.

Пусть R^n есть n-мерное евклидово пространство. Рассмотрим класс K линейных регрессий вида

$$x_0 = f(\tilde{x}) = \sum_{j=1}^{n} c_j x_j + c_{n+1}, \qquad (1)$$

у которых вектор аргументов $\tilde{x} = (x_1,\dots,x_n) \in R^n$ и вектор коэффициентов $(c_1,\dots,c_{n+1}) \in R^{n+1}$. Сопоставим конкретной регрессии $f \in K$ число $s(f) = |\{c_j \neq 0\}|$. Назовем его сложностью регрессии f.

Имеется обучающая выборка, состоящая из m объектов. Информация об этих объектах собрана в таблицу $T = (\tau_{ij})$, $i = 1,\dots,m$, $j = 0,1,\dots,n$. Элемент τ_{ij} есть значение переменной x_j на i-м объекте. Нас будут интересовать те регрессии из класса K, которые достаточно хорошо аппроксимируют зависимость x_0 от $x_1,\dots,x_n$, описываемую таблицей T. За качество аппроксимации, обеспечиваемое регрессией $f \in K$, примем величину

$$\Phi(f) = \| \sum_{j=1}^{n} c_j \tilde{\tau}_j + c_{n+1} \tilde{e} - \tilde{\tau}_0 \|, \qquad (2)$$

где вектор $\tilde{\tau}_j = (\tau_{1j},\dots,\tau_{mj}) \in R^m$, $\|\tilde{\tau}_j\| = \sum_{i=1}^{m} \tau_{ij}^2$ и компоненты m-мерного вектора $\tilde{e}$ есть единицы.

ЗАДАЧА. Даны таблица T и число $\varepsilon' \in [0,1]$, требуется найти регрессию $f \in K$ такую, что

$$s(f) \to \min_{f \in K}, \qquad (3)$$

$$\phi(f) \le \varepsilon = \varepsilon' \|\tilde{\tau}_0\|, \qquad (4)$$

либо установить, что множество таких регрессий пусто.

Отличие этой постановки от классической [1] заключается в том, что нужно найти регрессию минимальной сложности заданного качества.

§2. Описание алгоритма

Частным случаем задачи (3)-(4) является задача поиска минимального по весу решения системы линейных уравнений [2, с.315], которая NP-полна. Это не оставляет надежд на нахождение экономичного алгоритма, дающего точное решение задачи (3)-(4). Остается лишь одна возможность - решать приближенно. Ниже излагается алгоритм поиска регрессии $f \in K$, который в случае существования решения задачи (3)-(4) обеспечивает соблюдение требования (4), но не гарантирует выполнения требования (3). Алгоритм осуществляет перебор конечного числа g регрессий, удовлетворяющих неравенству (4). По числу операций умножения его трудоемкость равна $O(g \cdot n^3)$, что в n раз меньше, чем трудоемкость алгоритма, предложенного в [3]. Опишем схему алгоритма.

В работе [3] задача (3)-(4) сводится к следующей задаче булева программирования. Требуется отыскать бинарный набор $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_{n+1}) \in B^{n+1}$, удовлетворяющий условиям

$$|\tilde{\alpha}| = \sum_{j=1}^{n+1} \alpha_j \to \min_{\tilde{\alpha} \in B^{n+1}},$$

$$\varphi(\tilde{\alpha}) = \min_{(c_1, \ldots, c_{n+1}) \in R^{n+1}} \Big\| \sum_{j=1}^{n} \alpha_j c_j \tilde{\tau}_j + \alpha_{n+1} c_{n+1} \tilde{e} - \tilde{\tau}_0 \Big\| \le \varepsilon, \qquad (6)$$

где B^{n+1} есть множество $(n+1)$-мерных бинарных наборов $\tilde{\alpha}$. Число $|\tilde{\alpha}|$ называется нормой набора $\tilde{\alpha}$.

Будем говорить, что набор $\tilde{\beta} = (\beta_1, \ldots, \beta_{n+1})$ предшествует набору $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_{n+1})$, если для всех $j = 1, \ldots, n+1$ выполняется неравенство $\beta_j \le \alpha_j$. Отношение предшествования обозначим значком $\tilde{\beta} \prec \tilde{\alpha}$.

Пусть $H(\tilde{\alpha})$ есть множество наборов $\tilde{\beta}$ таких, что $|\tilde{\beta}| = |\tilde{\alpha}|-1$ и $\tilde{\beta} \not\downarrow \tilde{\alpha}$. Заметим, что если $\tilde{\beta} \not\downarrow \tilde{\alpha}$, то $\varphi(\tilde{\beta}) \geq \varphi(\tilde{\alpha})$. Набор $\tilde{\alpha}$ назовем тупиковым, если $\varphi(\tilde{\alpha}) \leq \varepsilon$ и для всех $\tilde{\beta} \in H(\tilde{\alpha})$ справедливо неравенство $\varphi(\tilde{\beta}) > \varepsilon$.

Точное решение задачи (5)-(6) является тупиковым набором. В качестве приближенного ее решения можно брать любой тупиковый набор. Алгоритм находит несколько тупиковых наборов $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_g$, и среди них выбирается тот, норма которого минимальна.

Список тупиковых наборов $S = \{\tilde{\alpha}_1, \ldots, \tilde{\alpha}_g\}$ формируется в процессе работы алгоритма. Вначале он пуст. Допустим, что алгоритм наработал список $S = \{\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t\}$. Назовем наборы $\tilde{\alpha}$ и $\tilde{\beta}$ сравнимыми, если выполняется одно из соотношений: $\tilde{\alpha} \not\downarrow \tilde{\beta}$ или $\tilde{\beta} \not\downarrow \tilde{\alpha}$. Обозначим через $C(\tilde{\alpha})$ множество наборов $\tilde{\beta}$, сравнимых с набором $\tilde{\alpha}$. Образуем множество $C = \bigcup_{\alpha \in S} C(\tilde{\alpha})$. Для всех $\tilde{\beta} \in C$ значение предиката $\varphi(\tilde{\beta}) \leq \varepsilon$ уже известно. С помощью процедуры ПОКРЫТИЕ алгоритм выбирает набор $\tilde{\gamma} \in B^{n+1} \setminus C$. Затем вычисляется значение $\varphi(\tilde{\gamma})$. Если $\varphi(\tilde{\gamma}) \leq \varepsilon$, то существует тупиковый набор $\tilde{\beta} \not\downarrow \tilde{\gamma}$. Алгоритм разыскивает $\tilde{\beta} \not\downarrow \tilde{\gamma}$, используя процесс пошаговой регрессии [4,с.367-370], реализованный в процедуре РЕГРЕССИЯ. Далее тупиковому набору $\tilde{\beta}$ присваивается имя $\tilde{\alpha}_{t+1}$, и он заносится в список S.

На первом шаге, когда $S = \emptyset$, в качестве исходного $\tilde{\gamma}$ берется набор, все компоненты которого единицы. Если $\varphi(\tilde{\gamma}) > \varepsilon$, то для данной таблицы T и заданного значения ε решения задачи (5)-(6) не существует. Алгоритм заканчивает работу, если найдено g тупиковых наборов или если процедура ПОКРЫТИЕ не может отыскать набор $\tilde{\gamma} \in B^{n+1} \setminus C$ со свойством $\varphi(\tilde{\gamma}) \leq \varepsilon$.

§3. Описание основных процедур

Процедура ПОКРЫТИЕ $(\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t)$. Построим бинарную таблицу

$$G = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1\,n+1} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_{t1} & \alpha_{t2} & \cdots & \alpha_{t\,n+1} \end{bmatrix},$$

где $\tilde{\alpha}_i = (\alpha_{i1}\ \alpha_{i2}, \ldots, \alpha_{in+1})$. Набор столбцов $\nu_1, \ldots, \nu_s$ таблицы G назовем покрытием, если в образованной им подтаблице в каждой строке есть по крайней мере одна единица. Покрытие является тупиковым, если любой его собственный поднабор столбцов покрытием не

является. Сопоставим тупиковому покрытию $\nu_1,\dots,\nu_s$ бинарный набор $\tilde{\gamma} = (\gamma_1,\dots,\gamma_{n+1})$, у которого $\gamma_i = 0$, если $i \in \{\nu_1,\dots,\nu_s\}$, и $\gamma_i = 1$, если $i \notin \{\nu_1,\dots,\nu_s\}$. Легко показать, что построенный таким способом набор $\tilde{\gamma} \notin C$.

Процедура РЕГРЕССИЯ ($\tilde{\gamma}$). Входным параметром процедуры является бинарный набор $\tilde{\gamma}$, полученный в результате работы процедуры ПОКРЫТИЕ. Процедура реализует процесс пошаговой регрессии с использованием так называемой операции "выметания" для включения регрессоров.

Определим операцию "выметания" (см. [4, с.341]). Пусть T_* есть матрица, составленная из столбцов $\tilde{\tau}_1,\tilde{\tau}_2,\dots,\tilde{\tau}_n,\tilde{\tau}_{n+1} = \tilde{e}, \tilde{\tau}_0$; A – матрица Грамма, определяемая равенством $A = T_*^T T_* = (a_{ij})_{(n+2)\times(n+2)}$. Операцией "выметания" по k-у ($k < n+2$) параметру называется следующее преобразование матрицы A в матрицу $A^* = (a^*_{ij})_{(n+2)\times(n+2)}$:

$$a^*_{kk} = \frac{1}{a_{kk}}, \quad a^*_{ik} = -\frac{a_{ik}}{a_{kk}} \ (i \neq k),$$

$$a^*_{kj} = \frac{a_{kj}}{a_{kk}} \ (j \neq k), \quad a^*_{ij} = a_{ij} - \frac{a_{ik}\cdot a_{kj}}{a_{kk}} \ (i,j \neq k). \tag{7}$$

Пусть к исходной матрице A была применена операция "выметания" по параметрам с номерами $z_1,z_2,\dots,z_r$ (все z_i различны) и в результате получена матрица A^*. Возьмем бинарный набор $\tilde{\alpha}^*$ такой, что $\alpha^*_i = 1$, если $i \in Z = \{z_1,\dots,z_r\}$, и $\alpha^*_i = 0$, если $i \notin Z$. Тогда в матрице A^* элемент

$$a^*_{n+2,n+2} = \varphi(\tilde{\alpha}^*) = \min_{(c_1,\dots,c_{n+1})\in R^{n+1}} \Big\| \sum_{j=1}^{n+1} \alpha^*_j c_j \tilde{\tau}_j - \tilde{\tau}_0 \Big\|. \tag{8}$$

Кроме того, $a^*_{j,n+2} = c_j$ для $j \in Z$. Таким образом используя операцию "выметания", можно одновременно получить $\varphi(\tilde{\alpha}^*)$ и сами оптимальные коэффициенты регрессии.

Схема работы процедуры. Пусть $\tilde{\gamma} = (\gamma_1,\dots,\gamma_{n+1})$. Образуем множество Z, в которое включаются все номера i такие, что $\gamma_i = 1$. Пусть $Z = \{z_1,\dots,z_p\}$. Положим $\beta_i = 0$ для всех $i = 1,\dots,n+1$. Среди элементов множества Z выбираем такой, на котором достигается $\max\limits_{k\in Z} \frac{a_{n+2,k}\cdot a_{k,n+2}}{a_{kk}}$, т.е. максимально уменьшается значение эле-

мента $a_{n+2,n+2}$ (см. (7),(8)). Полагаем $\beta_k = 1$ и проводим операцию "выметания" по параметру k. В полученной матрице A^* анализируется элемент $a^*_{n+2,n+2}$. Если его значение меньше ϵ, то полученный набор $\tilde{\beta} = (\beta_1, \ldots, \beta_{n+1})$ является тупиковым, а соответствующие ему оптимальные коэффициенты определяются соотношением: $c_j = a^*_{j,n+2}$, если $\beta_j = 1$, и $c_j = 0$, если $\beta_j = 0$. Если $a^*_{n+2,n+2} > \epsilon$, то полагаем $Z = Z \setminus \{k\}$ и повторяем описанные выше действия с новым Z.

Возможно, что после нескольких шагов среди элементов множества Z окажется такой номер i, что $\frac{a_{ii}}{(\tilde{\tau}_i, \tilde{\tau}_i)} < \delta$, где δ – число, близкое к нулю, тогда i исключается из множества Z. Эта операция нужна для того, чтобы избежать появления в регрессии почти линейно-зависимых переменных.

Из описания схемы работы алгоритма и процедуры РЕГРЕССИЯ видно, что для нахождения одного тупикового набора требуется $O(n^2 \cdot s)$ операций умножения, где s – сложность полученной регрессии. Таким образом, трудоемкость описанного выше алгоритма можно оценить как $O(g \cdot n^3)$, где g – число найденных регрессий.

В заключение отметим, что на основе данного алгоритма написаны и работают программы построения регрессионных и авторегрессионных моделей прогноза [5].

Литература

1. ДЕМИДЕНКО Е.З. Линейная и нелинейная регрессии. –М.: Финансы и статистика, 1981. – 302 с.

2. ГЭРИ М., ДЖОНСОН Д. Вычислительные машины и труднорешаемые задачи. –М.: Мир, 1982. – 416 с.

3. УСТЮЖАНИНОВ В.Г. Модели прогноза минимальной сложности. –В кн.: Анализ разнотипных данных (Вычислительные системы, вып. 99). Новосибирск, 1983, с. 88–101.

4. СЕБЕР Дж. Линейный регрессионный анализ. –М.: Мир, 1980. – 456 с.

5. УСТЮЖАНИНОВ В.Г., ШЕМЯКИНА Е.Н. Пакет прикладных программ "СИНТЕЗ" для построения и анализа моделей прогноза. –В кн.: Методы анализа данных (Вычислительные системы, вып. III). Новосибирск, 1985, с. 77–89.

6. ЛБОВ Г.С. Алгоритмы выбора эффективной системы признаков. –В кн.: Распознавание образов в социальных исследованиях. Новосибирск, 1968, с. 148–159.

Поступила в ред.-изд.отд.
10 сентября 1985 года