АНАЛИЗ ДАННЫХ И ЗНАНИЙ В ЭКСПЕРТНЫХ СИСТЕМАХ
(Вычислительные системы)
1990 Выпуск 134

УДК 519.237

ОЦЕНИВАНИЕ ОШИБКИ ПРОГНОЗА ПРИ ЗАПОЛНЕНИИ ПРОБЕЛОВ В ЭМПИРИЧЕСКИХ ТАБЛИЦАХ ДАННЫХ

Г.В. Ульянов

В работах [1,2] было предложено семейство алгоритмов ZL, предназначенных для заполнения пробелов и редактирования элементов в эмпирических таблицах данных. Экспериментальное сравнение алгоритмов по средней ошибке прогноза показало, что в среднем ошибка прогноза на рассмотренных данных получается вполне удовлетворительной; были указаны и наиболее перспективные в этом смысле алгоритмы семейства. Однако разброс ошибок велик - от 0 до 500%, и пользователь может усомниться в способности алгоритмов предсказывать значения элементов таблицы, а также в результатах сравнения и рекомендациях,так как не может заранее знать, относится ли какой-нибудь конкретный пробел в его данных к тому самому случаю с ошибкой в 500%. Очевидно, прогноз пробела должен сопровождаться оценкой ошибки и точности оценивания этой ошибки должно уделяться не меньше внимания, чем точности самого прогноза. В настоящей статье делается сравнение алгоритмов по эффективности оценивания ошибки прогноза, а также предлагаются советы по выбору той или иной оценки и по улучшению при помощи различных приемов точности прогнозов.

§1. Заполнение пробелов и оценивание ошибки. Введение

Пусть Z - двухвходовая таблица данных $(m \times n)$-типа "объект-свойство". Все, о чем пойдет речь далее, можно легко обобщить и на трехвходовые таблицы данных. Обозначим пробелы в таблице Z каким-нибудь числом PROB, отличным от всех остальных элементов таблицы. Введем обозначение для упорядоченного множества координат элементов Z :

$$I = (\tilde{I}, \prec) = \{(1,1),(1,2),\ldots,(m,n)\},$$

где $\tilde{I} = [1,m] \times [1,n]$- целочисленный прямоугольник, а "$\prec$" - лексикографический порядок. Будем записывать элементы z_{ij} таблицы Z через z_i, где $i \in I$ - вектор координат элемента $i = (1,j)$; множество координат пробелов обозначим через I^-: $\forall i \in I^-$, z_i = PROB. Тогда $I = I^- \cup I^+$, где I^+ - множество координат комплектных элементов.

Будем считать, что для каждого элемента z_i, $i \in I$, алгоритм может вычислить прогноз $\hat{z}_i$. Если по каким-то причинам алгоритм выдает отказ от прогноза, то можно определить $\hat{z}_i$ = = PROB.

В идеале мы хотели бы иметь прогноз пробела с заданной точностью и надежностью. Этим целям в какой-то степени может служить приближенный доверительный интервал с надежностью p % для истинного значения элемента z_i :

$$z_i = \hat{z}_i \pm \hat{\Delta}_i \cdot t_{(p,m_1)}, \qquad (1)$$

где $\hat{\Delta}_i$ - предсказанная ошибка, а $t_{(p,m_1)}$ - $(100-p)/2$-процентиль t-распределения с $(m_1 - 1)$ степенями свободы, m_1 - объем выборки (количество строк в подматрице без предсказываемой, см. §2). Значения $t_{(p,m_1)}$ для некоторых m_1 и p заданы в табл. 1.

Таблица 1

Значения $t_{(p,m_1)}$

Надежность p	Объем выборки, m_1						
	8	10	15	20	25	30	50
90%	1.90	1.83	1.76	1.73	1.71	1.70	1.68
95%	2.37	2.26	2.15	2.09	2.06	2.05	2.01

Приближенность интервала заключается в предположении нормальности $(z_i - \hat{z}_i)$ и в несмещенности прогноза, т.е. $E(z_i - \hat{z}_i) = 0$, где E - знак матожидания. *Предсказанная ошибка* $\hat{\Delta}_i$ в формуле (1) должна тогда равняться $\hat{\Delta}_i = \hat{Sd}(\hat{z}_i) = \hat{Var}^{1/2}(\hat{z}_i)$ - стандартной ошибке прогноза, Var - знак дисперсии. Если прогноз не смещен, то $\hat{\Delta}_i = \hat{Sd}(\hat{z}_i) = (\hat{E}(z_i - \hat{z}_i)^2)^{1/2}$. Последний член равенства есть оценка среднеквадратической ошибки прогноза. Когда прогноз смещен, то равенство не выполняется и тогда эта величина оценивает Δ_i лучше, чем $\hat{Sd}(\hat{z}_i)$. О том, как следует вычислять Δ_i, подробно говорится в §3-4.

Очевидно, что $\hat{\Delta}_i$ может служить оценкой не только для величин типа $Sd(\hat{y}_i)$, значения которых мы на практике вычислить не можем, но и величины $\Delta_i = |z_i - \hat{z}_i|$, которая легко вычисляется. Назовем ее *фактической ошибкой* прогноза. Если $\hat{\Delta}_i$ хорошо оценивает Δ_i, то можно рассчитывать на то, что доверительный интервал будет накрывать истинное значение z_i с надежностью, близкой к p %.

Для того чтобы ошибки прогнозов для разных элементов таблицы были сравнимы, будем вычислять их в процентах к стандартным ошибкам соответствующих столбцов таблицы. Эти ошибки будем называть *квадратическими:* фактическая квадратическая ошибка прогноза $s_i = 100 \cdot (\Delta_i / \hat{Sd}_i)$ % и предсказанная квадратическая ошибка $\hat{s}_i = 100 \cdot (\hat{\Delta}_i / \hat{Sd}_i)$%, где $\hat{Sd}_i$ - оценка стандартной ошибки столбца, в котором находится элемент z_i. Вопрос о близости $\hat{s}_i$ и s_i будет обсуждаться в §5 и далее.

Теперь мы можем привести схему обработки одного элемента z_i алгоритмом ZL: $A(Z,i) = (\hat{z}_i, \hat{s}_i, s_i)$. В обработке обычно различают два основных режима: а) редактирование комплектных элементов и б) заполнение пробелов. Разница между ними лишь в том, что фактическая ошибка для пробела не может быть вычислена. Для $\forall i \in I$ s_i = PROB.

Приведем общую схему работы любого алгоритма заполнения-редактирования типа ZL:

$$A(Z, I_p, I_0) = (\hat{Z}(I_0), \hat{S}(I_0), S(I_0)).$$

Здесь Z - таблица данных; $I_p \subset I$ - множество координат предикторных элементов таблицы; $I_0 \subset I$, $I_0 = (i_1, \ldots, i_d)$, - множество координат обрабатываемых элементов таблицы; $\hat{Z}(I_0) = (\hat{z}_{i_1}, \ldots, \hat{z}_{i_d})$ - вектор прогнозов для элементов $z_{i_1}, \ldots, z_{i_d}$; $S(I_0) = (s_{i_1}, \ldots, s_{i_d})$ - вектор фактических ошибок; $\hat{S}(I_0) = (\hat{s}_{i_1}, \ldots, \hat{s}_{i_d})$ - вектор предсказанных ошибок.

Элемент z_{i_0}, $i_0 \in I_p$, называется *предикторным,* если он используется для прогноза всех элементов $i \in I_0$, кроме самого элемента z_{i_0}. Обработка одного элемента z_i, $i \in I_0$, более полно определяется теперь как

$$A(Z,I_p\setminus\{i\},\{i\}) = (\{\hat{z}_i\},\{\hat{s}_i\},\{s_i\}).$$

Для простоты записи будем опускать фигурные скобки:

$$A(Z,I_p\setminus\{i\},i) = (\hat{z}_i,\hat{s}_i,s_i).$$

Все элементы $i \notin I_p$ при вычислении прогнозов и ошибок временно заменяются пробелами ("закрываются"). Непредикторные элементы $i \notin I_p$ делятся на пробелы и *"псевдопробелы"*; последние отличаются от пробелов только тем, что их редактируют, а не заполняют, т.е. для них можно вычислить фактическую ошибку (если $i \notin I_p$, $i \in I_0$, $i \in I^+$, то $s_i \neq$ PROB в общем случае).

Очевидно, что псевдопробелы в таблице организует сам пользователь. Зачем они нужны? С практической стороны эта конструкция полезна для нейтрализации отдельных элементов таблицы, подозрительных на аномальность. Если при обычном редактировании на первом прогоне программы обнаружены такие элементы, то при объявлении их псевдопробелами на втором прогоне, редактирование может выявить среди них фальшивые выбросы, настоящие выбросы при этом проявятся чётче, а среди остальных элементов могут обнаружиться другие выбросы, ранее замаскированные влиянием выявленных выбросов. Особо явно о выбросе говорит малая величина предсказанной и большая величина фактической ошибки.

С исследовательской точки зрения псевдопробелы можно эффективно использовать при экспериментальном изучении свойств алгоритмов заполнения пробелов, что и будет продемонстрировано в §7.

§2. Алгоритмы семейства ZL. Краткое описание

Подробное описание алгоритмов семейства ZL дано в [1]. Однако чтобы далее можно было говорить об оценках ошибки, дадим все же краткое изложение сути алгоритмов.

Все элементы таблицы обрабатываются (редактируются, заполняются) алгоритмом независимо друг от друга по одной и той же схеме. Пусть задана таблица данных $Z(m \times n)$. Алгоритмы семейства ZL требуют выполнения следующих шагов.

1. Отыскать очередной обрабатываемый элемент $z_{i_0} = z_{i_0 j_0}$.

2. Заполнить все пробелы в столбцах таблицы Z (кроме пробелов, находящихся в i_0-строке и j_0-столбце) средними $\bar{z}_j$ по столбцам (глобальными средними). Все характеристики столбцов вычисляются без учета элементов i_0-строки.

3. Нормировать столбцы по дисперсиям

$$z'_{ij} = (z_{ij} - \bar{z}_j) / \hat{Sd}(z_j).$$

4. Выбрать m_1 строк, ближайших к строке i_0 с точки зрения евклидова расстояния и не имеющих пробела в столбце j_0.

5. Построить матрицу выбранных строк $C(m_1 \times n)$, вычислить средние по столбцам матрицы (локальные средние) и заполнить этими средними пробелы в столбцах матрицы C (кроме пробелов в строке и столбце, в которых находится обрабатываемый элемент).

6. Подать эту матрицу на вход одного из алгоритмов регрессионного анализа, содержащего процедуру отбора предикторов. Этот алгоритм отберет заданное число n_1 (или меньше) наиболее информативных столбцов, не имеющих пробела в i_0-й строке, и оценит регрессию j_0-го столбца на n_1 отобранных столбцов-предикторов матрицы.

7. Применяя оцененную функцию регрессии к элементам i_0-й строки, вычислить прогноз $\hat{z}'_{i_0} = \hat{z}'_{i_0 j_0}$, обратить

для него нормировку $\hat{z}'_i \to \hat{z}_i$, а также найти предсказанную $\hat{s}_i$ и фактическую s_i ошибки (если $i \in I^+$).

8. Если есть еще необработанные элементы, вернуться к шагу 1, иначе стоп.

Нормировка столбцов предпринимается с целью обеспечения инвариантности алгоритма к допустимым преобразованиям шкал признаков. В данном случае *инвариантность* понимается в следующем смысле: от алгоритма A, такого что $A(Z) = \hat{Z}$, где $\hat{Z}(m \times n)$ - таблица прогнозов, требуется, чтобы $A(D(Z)) = D(A(Z))$, где $D = (d_1, \ldots, d_n)$ - вектор допустимых преобразований столбцов. Данные все измерены в количественных шкалах, поэтому преобразование каждого столбца z_j линейно: $d_j(z_j) = \beta_j z_j + \gamma_j$, где β_j, γ_j - константы. Нормировка N, вводимая как часть алгоритма, обеспечивает алгоритму инвариантность благодаря своему свойству $N_D(D(Z)) = N_Z(Z)$. Обозначения N_D, N_Z означают, что вид оператора нормировки зависит от нормируемой таблицы данных. Мы имеем

$$A(N_D(D(Z))) = A(N_Z(Z)),$$

$$N_Z^{-1}(A(N_D(D(Z)))) = N_Z^{-1}(A(N_Z(Z))).$$

Так как $N_D \circ D = N_Z$, то $D^{-1} \circ N_D^{-1} = N_Z^{-1}$. Таким образом, $D^{-1} \circ N_D^{-1} \circ A \circ N_D \circ D(Z) = N_Z^{-1} \circ A \circ N_Z(Z)$. Вводя нормировку в состав алгоритма, мы переходим фактически к алгоритму $\bar{A}$: $\bar{A}(Z) = N_Z^{-1} \circ A \circ N_Z(Z)$, который инвариантен

$$D^{-1}(\bar{A}(D(Z))) = \bar{A}(Z).$$

Далее вместо $\bar{A}$ будем везде писать просто A.

Семейство ZL состоит из 7 алгоритмов:

1) базовые: ZL-СТ, ZL-ПШ, ZL-НСПА;

2) смешанные: ZL-НПШ, ZL-НСТ;

3) составные: ZL-НСПА+СТ, ZL-НСПА+ПШ.

Базовые алгоритмы отличаются друг от друга моделями регрессии: линейной моделью (ZL-СТ, ZL-ПШ), непараметрической моделью (ZL-НСПА), а также методами отбора столбцов-предикторов: ZL-СТ (ступенчатый метод),ZL-ПШ (пошаговый метод),ZL-НСПА (метод СПА [3]).Все эти методы рассматривались в [1] и здесь затрагиваться не будут. Моделей же регрессии мы коснемся в §3-4.

В *смешанных* алгоритмах отбор предикторов осуществляется методами ПШ и СТ, а регрессия оценивается непараметрически. В *составных* алгоритмах для каждого элемента вычисляются два прогноза - параметрический и непараметрический,после чего определяется оптимальный прогноз по минимуму предсказанной ошибки.

§3. Оценивание ошибки прогноза в линейной регрессии

Отобранные строки и столбцы вместе со строкой l_0 и столбцом j_0 образуют предсказывающую подматрицу B $(m_1+1)\times(n_1+1)$. Без потери общности можно считать, что пробел $y_0 = z_{i_0}$ находится в левом верхнем углу матрицы B. Обозначим столбцы матрицы через $y, x_1, \ldots, x_{n_1}$, а строки - $b_i = (y_i, \underline{x}_i^T)$; $\underline{x}_i^T = (x_{i1}, \ldots, x_{in_1}), i = 0,1,\ldots,m_1$. Введем также строку средних по столбцам матрицы $B - (\bar{y}, \underline{\bar{x}}^T) = (\bar{y}, x_1, \ldots, x_{n_1})$:

	y	$x_1 \ldots x_{n_1}$	
b_0	y_0	$\underline{x}_0^T$	
b_1	y_1	$\underline{x}_1^T$	$= B$
$\ldots$	$\ldots$	$\ldots$	
b_{m_1}	y_{m_1}	$\underline{x}_{m_1}^T$	

Линейный прогноз основывается на линейной модели:

$$y_l = \bar{y} + \beta^T \cdot (\underline{x}_l - \bar{\underline{x}}) + \varepsilon_l ; \quad l = 0,1,\ldots,m_1,$$

где β - вектор параметров длины n_1, а ε - вектор случайных ошибок длины m_1. При этом предполагается, что $(l = 0,1,\ldots,m_1)$:

$$E\varepsilon_l = 0; \quad Var\varepsilon_l = \sigma^2; \quad E(\varepsilon_l \varepsilon_k) = 0, \quad \forall l \neq k.$$

Тогда прогноз $\hat{y}_0$ элемента y_0 вместе с прогнозами элементов y_l, $l = 1,\ldots,m_1$, в алгоритме ZL-ПШ вычисляется следующим образом:

$$\hat{y}_l = \bar{y} + \hat{\beta}^T \cdot (\underline{x}_l - \bar{\underline{x}}); \quad l = 0,1,\ldots,m_1,$$

где $\hat{\beta} = \hat{\Sigma}_x^{-1} \hat{\Sigma}_{xy}$, а $\hat{\Sigma}_x$ - оценка ковариационной матрицы переменных $x_1,\ldots,x_{n_1}$, $\hat{\Sigma}_{xy}$ - оценка вектора ковариаций между величиной y и переменными $x_1,\ldots,x_{n_1}$. Определим

$$r^2(\underline{x}_l, \bar{\underline{x}}) = (\underline{x}_l - \bar{\underline{x}})^T \hat{\Sigma}_x^{-1} (\underline{x}_l - \bar{\underline{x}}), \quad l = 0,1\ldots, m_1,$$

- квадрат расстояния Махалонобиса между строкой $\underline{x}_l^T$ и строкой средних $\bar{\underline{x}}^T$ в пространстве предикторов $x_1,\ldots,x_{n_1}$. Величина

$$h_l = (1/m_1) + r^2(\underline{x}_l, \bar{\underline{x}})/(m_1 - 1), \quad (1/m_1) \leq h_l < 1,$$

называется величиной разбалансировки l-го наблюдения [4, с.163]. Если $h_l > 2(n_1 + 1)/m_1$ (см. [5]), то эта точка $\underline{x}_l$ относится к точкам сильной разбалансировки (leverage point), т.е. можно считать, что она является выбросом в пространстве предикторов. В частности, если это касается точки $\underline{x}_0$,

то нельзя ожидать хорошего прогноза (в смысле малой ошибки) для y_0 .

Теперь можно определить оценки для ошибки прогноза (в скобках указан соответствующий режим алгоритма ZL-ПШ):

1. Среднеквадратическая ошибка (ISHOS=0):

$$\Delta_0^2 = (m_1 - n_1 - 1)^{-1} \sum_{i=1}^{m_1} (y_i - \hat{y}_i)^2 .$$

2. Оценка дисперсии прогноза (ISHOS=1): $\Delta_1^2 = \Delta_0^2 h_0$.

3. Оценка ошибки по методу кросс-проверки (cross-validation) [7,8] (ISHOS=2):

$$\Delta_2^2 = \frac{1}{m_1 - n_1 - 1} \sum_{i=1}^{m_1} (y_i - \hat{y}_{i(i)})^2 =$$

$$= \frac{1}{m_1 - n_1 - 1} \sum_{i=1}^{m_1} \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2 .$$

Здесь прогноз каждого элемента y_i вычисляется без использования в формулах i-й строки (leave-one-out):

$$\hat{y}_{i(i)} = \bar{y}_{(i)} + \hat{\beta}^T_{(i)} (\underline{x}_i - \underline{\bar{x}}_{(i)}); \quad \hat{\beta}_{(i)} = \hat{\Sigma}^{-1}_{x(i)} \hat{\Sigma}_{xy(i)} .$$

4. Оценка дисперсии прогноза по методу "кросс-проверки" (ISHOS=3): $\Delta_3^2 = \Delta_2^2 h_0$.

Первые две оценки являются классическими, а Δ_0 - оценка ошибки безусловной, усредненной по всем точкам. Обычно в линейной регрессии рекомендуется использовать Δ_1 , так как она учитывает "типичность" точки, в которой делается прогноз по отношению к остальным, т.е. величину разбалансировки. В [6,с.89] указано, что в рамках линейной модели с нормальными ошибками

доверительный интервал, основанный на Δ_1, является несмещенным и равномерно наиболее точным. В случае умеренного отклонения данных от линейной модели обычно советуют применять универсальный метод кросс-проверки - Δ_2, сглаживающий последствия такого несоответствия. Однако Δ_2 не учитывает "типичность" наблюдения. Предлагается в случае сильной неоднородности данных (расслоение, гетероскедастичность ошибок, умеренные выбросы) и одновременного отклонения самой модели от линейности использовать гибкую оценку Δ_3.

§4. Оценивание ошибки в непараметрической регрессии

Непараметрический прогноз основывается на непараметрическом оценивании функции регрессии $E(y|\underline{x}) = f(\underline{x})$ для модели $y_i = f(\underline{x}_i) + \epsilon_i$, $i = 0,1,\ldots, m_1$, где f - некоторая достаточно гладкая в общем случае нелинейная функция. Значения регрессии $f(\underline{x})$ в интересующих нас точках $\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{m_1}$ оцениваются по методу m_1-ближайших соседей с ядерными весами (подробнее см. в [1]). Прогноз элемента y_0 в этом случае равен

$$\hat{y}_0 = \hat{E}(y|\underline{x}_0) = \Big(\sum_i y_i w_{i0}\Big) \Big/ \sum_i w_{i0}, \quad i = 1,\ldots, m_1,$$

где $w_{ik} = \exp(-\|\underline{x}_i - \underline{x}_k\|^2/(2h^2))$ - веса пар строк, $i,k = 0,1,\ldots, m_1$; $\|\underline{x}_i - \underline{x}_k\|$ - евклидово расстояние между $\underline{x}_i$ и $\underline{x}_k$.

Прогноз зависит от параметра сглаживания h (bandwidth). Оптимальный прогноз должен минимизировать по h некоторую оценку ошибки прогноза $\hat{\Delta}(h)$. Ниже мы обсудим, что можно взять в качестве такого критерия оптимизации.

По аналогии с формулой для $\hat{E}(y|\underline{x}_0)$ можно построить выражение для "ошибки" - оценки условной дисперсии в точке прог-

ноза $\underline{x}_0$ (суммирование везде по $l=1,\ldots,m_1$):

$$\hat{c}^2 = \widehat{Var}(y,\underline{x}_0) = (m_0-1)^{-1}\sum_l ((y_l-\hat{y}_0)^2 w_{l0})/\sum_l w_{l0} =$$

$$= (m_0-1)^{-1}((\sum_l y_l^2 w_{l0})/(\sum_l w_{l0}) - \hat{y}_0^2).$$

Однако $\hat{c}$ - плохая оценка. Минимизировать $\hat{c}^2$ по h не имеет смысла, так как минимум достигается при $h = 0$ и в итоге получается примитивный прогноз по ближайшему соседу.

В отличие от линейного прогноза, непараметрический прогноз обычно имеет ненулевое смещение $\mathrm{Bias}(\hat{y}_0)=E(y_0-\hat{y}_0) \neq 0$ и доверительный интервал $y = \hat{y} \pm \hat{\Delta}\cdot t_{(p,m_1)}$, где $\hat{\Delta}^2 = \hat{E}(y_0-\hat{y}_0)^2$, будет иметь смещенный центр. Если бы было известно смещение, интервал можно было бы скорректировать:

$$y_0 = \hat{y}_0 + \mathrm{Bias}(\hat{y}_0) \pm (\hat{\Delta}^2 - \mathrm{Bias}^2(\hat{y}_0))^{1/2}\, t_{(p,m_1)}.$$

В противном случае для корректировки смещения следует применять какие-нибудь методы управления выборкой (resampling), например "складной нож" (jackknife) или бутстреп (bootstrap) [9]. Последний здесь рассматриваться не будет, так как некоторые алгоритмы семейства ZL сами,так же как и бутстреп, являются "компьютерно-интенсивными".

Пусть мы имеем джекнайф-оценки матожидания $\hat{E}\hat{y}_0$ и смещения $\mathrm{Bias}(\hat{y}_0)$ $(j=1,\ldots,m_1;\ k=0,1,\ldots,m_1)$:

$$\hat{E}\hat{y}_0 = m_1^{-1}\sum_{l=1}^{m_1}\hat{y}_{0(l)};$$

$$\widehat{\mathrm{Bias}}(\hat{y}_0) = (m_1-1)(\hat{y}_0 - \hat{E}\hat{y}_0);$$

$$\hat{y}_{k(l)} = (\sum_{j\neq l,k} y_j w_{jk})/\sum_{j\neq l,k} w_{jk};\quad l \neq k.$$

Тогда прогноз элемента y_0 с джекнайф-коррекцией смещения можно записать:

$$\tilde{y}_0 = \hat{y}_0 + \widehat{Bias}(\hat{y}_0) = m_1\hat{y}_0 - (m_1 - 1)\hat{E}\hat{y}_0 .$$

Здесь же можно привести статистику, которую обычно используют в качестве джекнайф-оценки как $Var\hat{y}_0$, так и $Var\,\tilde{y}_0$:

$$\widehat{Var}\,\hat{y}_0 = \widehat{Var}\,\tilde{y}_0 = \widehat{Var}(\hat{y}|x_0) = \frac{m_1 - 1}{m_1}\sum_1 (\hat{y}_{0(1)} - \hat{E}\hat{y}_0)^2 .$$

В качестве приближенного доверительного интервала для y_0 Тьюки [9,с.61] предлагал использовать $\tilde{y}_0 \pm t_{(p,m_1)} \cdot \hat{Sd}\hat{y}_0$, где $\hat{Sd}\,\hat{y}_0 = \widehat{Var}^{1/2} y_0$, однако Миллер [9] показал, что этот интервал очень груб и использование его неоправданно. Но сам прогноз $\tilde{y}_0$, тем не менее, представляет определенный интерес, и в §6 его изучение будет продолжено экспериментально.

По-видимому, если не замыкаться в рамках более или менее узких моделей, в настоящее время нет более надежных критериев, чем критерии, основанные на кросс-проверке (cross-validation) [7,8,10] (указаны наиболее ранние работы, в которых кросс-проверка применялась в регрессионном анализе). В одном из послед - них крупных обзоров методов статистического предсказания Рао [11] вполне определенно высказался в пользу преимуществ кросс-проверки как критерия при решении целого ряда сложных задач. Мы тоже будем строить свои оценки ошибки Δ на основе кросс-проверки.

Наиболее популярным критерием кросс-проверки является скользящий контроль (leave-one-out):

$$\hat{\Delta}_1^2 = \hat{E}(y - \hat{y})^2 = m_1^{-1} \sum_1 (y_1 - \hat{y}_{1(1)})^2 .$$

В нашем случае данную оценку можно дополнить оценкой этой ошибки в точке прогноза $\underline{x}_0$:

$$\hat{\Delta}_2^2 = \hat{E}(y_0 - \hat{y}_0)^2 = \hat{E}((y-\hat{y})^2 | \underline{x}_0) =$$

$$= \frac{\sum_1 (y_1 - \hat{y}_{1(1)})^2 w_{10}}{\sum_1 w_{10}}.$$

Оценка $\hat{\Delta}_2$ страдает теми же недостатками, что и $\hat{\sigma}$, и в качестве критерия использоваться не будет. Оценка $\hat{\Delta}_1$ слишком чувствительна к ошибкам на самых дальних соседях точки $\underline{x}_0$, поэтому обычно она хорошо работает как критерий оптимизации прогноза по h, но сильно переоценивает ошибку Δ. Для устранения этого эффекта была предложена [1] оценка по методу "медианной кросс-проверки":

$$\hat{\Delta}_3 = \underset{1}{\text{med}} |y_1 - \hat{y}_{1(1)}|; \quad 1 = 1, \ldots, m_1.$$

Таблица 2

Режимы алгоритма ZL-НСПА

IMA	Прогноз	Критерий	Оценка ошибки
0	y_0	$\hat{\Delta}_1$	$\hat{\Delta}_1$
1	$\tilde{y}_0$	$\hat{\Delta}_1$	$\hat{\Delta}_1$
2	y_0	$\hat{\Delta}_1$	$\hat{\Delta}_3$
3	$\tilde{y}_0$	$\hat{\Delta}_1$	$\hat{\Delta}_3$
4	y_0	$\hat{\Delta}_3$	$\hat{\Delta}_3$

Но из-за меньшей чувствительности к данным оценка $\hat{\Delta}_3$ хуже работает в качестве критерия оптимизации, чем $\hat{\Delta}_1$. Поэтому нам необходимо ввести в непараметрические алгоритмы некоторые гибридные режимы. Например, при IMA = 2 $\hat{\Delta}_1$ - критерий оптимизации по h, а после выбора оптимального прогноза $\hat{\Delta}_3$ - оценка ошибки последнего.

В программе, реализующей данный алгоритм, используются режимы, задаваемые параметром IMA и перечисленные в табл.2.

§5. Наборы данных. Обычное редактирование таблицы

Перейдем к экспериментальному изучению свойств оценок ошибки прогноза для различных алгоритмов и их режимов. Для этого мы располагаем шестью наборами данных различного характера. Четыре из них являются двухвходовыми таблицами куба данных (15x15x12), подробно описанного в [2]. Это - данные по 15 с/х показателям для 15 республик СССР за 12 лет (1970-81 гг.),взятые из статистического ежегодника "Народное хозяйство СССР" за соответствующие годы. В отличие от экспериментов, описанных в [2], исключен один показатель - валовый сбор картофеля,так как все алгоритмы по всем республикам и за все годы предсказывали его всегда с очень большой ошибкой. По-видимому, урожайность картофеля плохо предсказуема в принципе. Зато теперь в таблицы включены данные по РСФСР и УССР, которые на фоне данных по другим республикам выглядят как выбросы,так как теперь у нас есть средства борьбы с неоднородностью данных.

Итак, в эксперименте участвуют наборы данных №1 и 3: таблицы (15x15) типа "объект-свойство" - производство с/х продукции по республикам соответственно за 1975 г. и 1979 г.; набор данных №2: таблица (12x15) типа "время-свойство" - с/х показатели за ряд лет по Киргизской ССР; набор данных №4: таблица (15x12) типа "объект-время" - производство молока по республикам за ряд лет.

Кроме этих четырех наборов, привлекается набор данных №5, полученный путем моделирования и описанный подробно в [1]. Моделировалась линейная зависимость с нормальными ошибками и коэффициентом множественной корреляции $\rho = 0.7$ столбца-отклика таблицы со столбцами-предикторами, с 10% удалением элемен - тов таблицы. Удаление производилось случайно и повторялось са-

данное число раз (10). Прогнозировался элемент столбца-предиктора. На выход подавались вектор прогнозов этого элемента и вектор ошибок.

Набор данных №6: матрица (23x9), взятая из [12], где она использовалась в качестве тестового примера для проверки подпрограммы факторного анализа.

В ходе экспериментов нам хотелось выяснить, в каких режимах оценка $\hat{s}$ фактической квадратической ошибки s лучше оценивает s, т.е. когда распределение $e = s - \hat{s}$ с максимальной вероятностью сосредоточивается в окрестности нуля. Попутно нам придется сравнить алгоритмы и их режимы между собой и по точности прогноза, т.е. выяснить, для каких алгоритмов распределение самой фактической ошибки s с наибольшей вероятностью концентрируется возле нуля, хотя данному сравнению и была в какой-то степени посвящена предыдущая работа [1]. Очевидно, решать первую задачу, не затрагивая второй, невозможно, к тому же новые эксперименты внесли ряд корректировок в прошлое сравнение.

Первым экспериментом, в которых мы испытаем алгоритмы и их режимы, будет *обычное редактирование таблицы* Z $(m \times n)$ (см., например, [13, с.60]): $A(Z,I,I)=(\hat{Z}(I),\hat{S}(I),S(I))$. В этом эксперименте каждый элемент таблицы по очереди "закрывается" и предсказывается, после чего для него вычисляется фактическая ошибка. Пробелов, по сути, нет, но информация об элементе не используется при его прогнозировании: $A(Z, I\setminus\{i\}, i) = (\hat{z}_i, \hat{s}_i, s_i)$; $\forall i \in I$; $I^- = \emptyset$.

В ходе эксперимента образуются массивы фактических $\{s_i\}_{i\in I}$ и предсказанных $\{\hat{s}_i\}_{i\in I}$ ошибок соответственно; из них формируется массив разностей ошибок $e_i = s_i - \hat{s}_i$, $\{e_i\}_{i\in I}$ и модулей разностей ошибок $\{|e_i|\}_{i\in I}$.

Данные массива - это как бы выборки, по которым мы должны сравнивать распределения соответствующих величин s, e, $|e|$ для разных алгоритмов. С целью облегчения данного сравнения вводятся следующие выборочные характеристики - элементы дополненных пятичисловых сводок Тьюки [13] (назовем их условно *усредненными характеристиками*):

1) **Ql** - нижняя квартиль $(\alpha_{0.25})$;

2) **Me** - медиана $(\alpha_{0.50})$;

3) **Qu** - верхняя квартиль $(\alpha_{0.75})$;

4) **De** - верхняя дециль $(\alpha_{0.90})$;

5) **Int = (Qu-Ql)/2** - половина интерквартильного размаха;

6) $\bar{x}$ - среднее;

7) **Sd** $= \mathbf{Var}^{1/2}$ - стандартная ошибка.

Первые четыре характеристики - выборочные квантили; q-квантиль вариационного ряда $u_1, \ldots, u_k$ $(0<q<1)$ вычисляется как $\alpha_q = (u_{[qk+1]} + u_{]qk[})/2$, где $[v] \le v \le]v[$ означают целые числа, ближайшие к v снизу и сверху.

Таблица 3

Точность прогноза при редактировании набора данных №1.

В скобках значения параметра IMA

Алгоритм (Режим)	Фактическая квадратическая ошибка прогноза						
	Ql	Me	Qu	De	Int	$\bar{x}$	Sd
ПШ	2.7	7.2	35.4	106.6	16.3	40.3	92.8
НСПА(0,2)	3.2	8.0	40.2	124.7	18.5	50.3	112.4
НСПА(1,3)	3.5	9.3	41.1	158.7	18.8	54.2	115.6
НСПА(4)	2.6	8.6	37.0	121.2	17.2	49.3	111.9
СТ	2.5	6.5	25.5	112.6	11.5	37.2	84.5
НПШ(4)	3.0	8.5	36.0	121.3	16.5	49.6	109.1
НСТ(4)	2.8	9.4	37.2	121.3	17.2	48.0	103.6
НСПА+ПШ(4)	2.7	7.8	35.0	111.0	16.1	45.9	104.9
НСПА+СТ(4)	2.3	6.4	25.6	104.3	11.7	36.6	84.5
ЛСР	6.1	16.3	87.0	354.8	40.4	103.3	198.2

Таблица 4

Эффективность оценивания ошибки при редактировании набора данных №1. В скобках - значение параметра ISHOS для алгоритма ZL-ПШ и значение параметра IMA для остальных алгоритмов

Алгоритм (Режим)	Модуль разности ошибок прогноза					Разность ошибок	
	Q1	Me	Qu	De	Int	Me	Int
ПШ(0)	1.7	6.0	23.8	76.7	11.0	4.0	10.0
ПШ(1)	1.9	6.3	24.3	81.7	11.2	4.4	11.0
ПШ(2)	1.8	5.0	22.7	79.4	10.5	1.4	5.9
ПШ(3)	1.9	4.8	20.0	72.0	9.1	2.3	5.4
НСПА(0)	4.1	11.2	30.9	75.4	13.4	-4.5	9.5
НСПА(1)	3.8	10.4	31.1	82.7	13.6	-2.9	9.4
НСПА(2)	1.9	4.9	24.9	104.9	11,5	1.4	10.1
НСПА(3)	2.3	5.7	26.3	133.1	12.0	2.5	9.7
НСПА(4)	1.5	5.2	22.6	93.5	10.6	2.5	8.6
СТ	1.6	5.2	20.1	85.1	9.3	2.7	9.3
НПШ(4)	1.6	6.0	24.6	107.6	11.5	3.3	11.3
НСТ(4)	1.9	6.1	29.1	107.6	13.6	3.7	12.6
НСПА+ПШ(4)	1.6	4.7	20.5	85.1	9.4	1.8	8.4
НСПА+СТ(4)	1.5	4.7	17.9	78.5	8.2	1.5	6.7
ЛСР	4.3	10.4	39.4	265.5	17.5	1.9	20.2

Для изучения близости s к нулю мы будем применять все семь характеристик, а для исследования близости $s-\hat{s}$ к нулю часть характеристик приложим к $|e| = |s-\hat{s}|$, а часть (Me, Int) - к e, так как нас интересует не только близость, но также величина и знак смещения, чтобы знать, переоцениваем или недооцениваем мы ошибку.

Выводы из результатов редактирования всех шести наборов данных будут сделаны в §6. Сами же результаты полностью привести невозможно, поэтому мы поступим следующим образом. Редактирование одного набора данных (№1) будет освещаться довольно подробно для иллюстрации. В табл. 3 алгоритмы сравниваются по точности прогноза (по фактической квадратической ошибке s); в

табл.4 те же алгоритмы,а также различные их режимы сравниваются по эффективности оценивания ошибки (по $s-\hat{s}$ и $|s-\hat{s}|$). Кроме алгоритмов семейства ZL, в сравнении участвует один из простейших алгоритмов заполнения пробелов - локальное среднее по столбцу (ЛСР), т.е. среднее по столбцу матрицы B, в котором находится обрабатываемый элемент. Для смешанных и составных алгоритмов используются режимы IMA = 4 и ISHOS = 3 (при этом сочетании результаты наилучшие).

Далее,в табл.5 представлены результаты сравнения прогнозов и ошибок прогнозов для набора данных №2. Оставлены только базовые алгоритмы и наиболее результативные режимы.Эти два набора - №1 и №2 - будут в дальнейшем иллюстрировать различные приемы улучшения точности прогноза. В табл.6 соединены результаты сравнения прогнозов на наборах данных №5 и №6 для базовых алгоритмов. Эта таблица пригодится для иллюстрации выводов §6.

При обработке всех наборов данных использованы предсказывающие подматрицы (8x6), за исключением №5 (21x6).

§6. Результаты редактирования

На основании результатов экспериментов по редактированию таблиц можно сделать ряд выводов. Если сравнивать алгоритмы по *точности прогноза*, то обнаруживаются следующие закономерности.

1. Лучше всех элементы таблиц предсказывают самые трудоемкие составные алгоритмы (особенно ZL-НСПА+СТ), причем они, как правило, улучшают результаты как параметрических, так и соответствующих непараметрических алгоритмов.

2. Из базовых алгоритмов на первых четырех наборах (с/х данных) лучше работали параметрические алгоритмы (ZL-СТ и ZL-ПШ). По-видимому, это объясняется наличием линейных зависимостей в кубе данных, может быть, даже простых, так как часто побеждает алгоритм ZL-СТ.

Таблица 5

Редактирование набора данных №2

Алгоритм (Режим)	Фактическая квадратическая ошибка прогноза						
	Ql	Me	Qu	De	Int	$\bar{x}$	Sd
ПШ	12.1	27.1	61.6	106.1	24.8	44.4	48.6
НСПА(2)	17.7	35.0	61.6	98.6	21.9	46.3	49.1
НСПА(4)	19.0	32.2	61.6	96.1	21.3	46.9	49.8
СТ	13.6	27.9	60.2	114.0	23.3	45.9	49.5
Алгоритм (Режим)	Модуль разности ошибок прогноза					Разность ошибок	
	Ql	Me	Qu	De	Int	Me	Sd
ПШ(2)	8.2	15.6	36.0	61.0	13.9	4.0	18.3
ПШ(3)	6.9	17.3	38.4	74.1	15.8	11.3	18.9
НСПА(2)	8.7	19.6	37.9	57.2	14.6	-3.1	20.0
НСПА(4)	7.8	20.4	39.1	61.0	15.6	4.7	23.0
СТ	6.1	18.4	38.0	80.7	15.9	12.4	19.0

Таблица 6

Редактирование наборов данных №5 и 6

Алгоритм (Режим)	Фактическая квадратическая ошибка прогноза						
	Ql	Me	Qu	De	Int	$\bar{x}$	Sd
Набор №5							
ПШ	23.8	40.8	65.9	115.8	21.1	51.7	41.3
НСПА(2)	11.7	22.8	54.8	113.3	21.6	38.3	37.5
НСПА(4)	14.9	30.7	63.5	116.0	24.3	46.3	41.7
СТ	27.9	54.3	86.6	115.1	29.4	59.6	37.0
Набор №6							
ПШ	30.7	63.3	106.4	184.1	37.9	83.4	79.9
НСПА(2)	21.9	45.9	81.9	137.7	30.0	64.5	57.8
НСПА(4)	21.7	48.2	83.7	134.6	34.3	63.5	61.0
СТ	28.9	62.8	116.7	235.8	43.9	2153	8760

3. На двух других наборах (№5,6) победил ZL-НСПА. Там,где он оказался на высоте,следом шли составные,потом смешанные алгоритмы, а затем только ZL-СТ и ZL-ПШ. Возможно, здесь сказалась нелинейность зависимостей, а может, и бóльший объем выборки.

4. Смешанные алгоритмы предсказывают ненамного хуже, чем ZL-НСПА, а часто и лучше. В целом ZL-НСПА не очень оправдывает свою трудоемкость. Сравните затраченное время на редактирова - ние 225 элементов таблицы набора данных №1:

ЛСР - 50 с,

СТ - 55 с,

ПШ - 1 мин 50 с,

НСПА - 8 мин 56 с,

НПШ - 2 мин 51 с,

НСТ - 1 мин 55 с,

НСПА+ПШ - 10 мин 35 с,

НСПА+СТ - 10 мин 02 с.

5. Линейный прогноз не зависит от значений параметра ISHOS, но в ZL-НСПА прогноз зависит от IMA. Лучше всего прогнозируют режимы IMA = 2 и 4, они успешно конкурируют друг с другом: если режим IMA = 4 улучшил самые плохие и самые хорошие прогнозы по сравнению с IMA = 2, то зато ухудшил прогнозы средней точности, о чем говорит значение медианы и верхней квартили фактической ошибки.

6. Алгоритм ЛСР предсказывает намного хуже алгоритмов ZL, что говорит о способности последних оценивать зависимости.

При сравнении алгоритмов и их режимов по *эффективности оценивания ошибки* прогноза сделаны следующие выводы.

1. ZL-ПШ. Результаты в целом подтверждают рекомендации, данные в §3. Режим ISHOS = 0 нигде не дал хороших оценок ошибки. Режим ISHOS = 1 лишь для набора №5 (разделив успехи с режимом ISHOS = 3) дал хорошие результаты, т.е. там, где дейст -

вительно моделировались линейные зависимости с нормальными ошибками. На двух других наборах (№2,4) сравнительно однородных данных победил режим ISHOS = 2. На трех оставшихся, самых сложных наилучшие результаты показал режим ISHOS = 3. В целом при отсутствии какой-либо информации о данных следует рекомендовать наиболее гибкую оценку ошибки, соответствующую режиму ISHOS = 3.

Оценка ошибки при ISHOS = 2 имеет наименьшее смещение. В то же время $\hat{s}$ при всех режимах недооценивает ошибку s. Видимо, дело в том [6,с.89], что в оценках $\hat{\Delta}_1$ и $\hat{\Delta}_3$ учитывается лишь дисперсия прогноза Var $\hat{y}_0$, но не учитывается дисперсия самого наблюдения Var y_0, т.е. доверительный интервал строится для ожидаемого значения элемента Ey_0, а не для самого элемента $y_0 = Ey_0 + \varepsilon_0$. С учетом этих соображений следовало бы в формулах для $\hat{\Delta}_1$ и $\hat{\Delta}_3$ заменить множитель h_0 на $(h_0 + 1)$. Например, $\hat{\Delta}_3^2 = \hat{\Delta}_2^2(h_0 + 1)$.

2. ZL-НСПА. В данном случае основная борьба разгорается между режимами IMA = 2 и 4. Режим IMA = 2 по сравнению с режимом IMA = 4 систематически улучшает плохие прогнозы и несколько ухудшает хорошие. Это выражается в том, что верхние квартиль и дециль величины $|e| = |s - \hat{s}|$ для режима IMA = 2 обычно меньше (наборы №2,3,4,6), а нижняя квартиль $|e|$ меньше уже для режима IMA = 4 (наборы №1-6). В целом рекомендуется режим IMA = 2, так как он наиболее успешно работает в наиболее неблагоприятных ситуациях. Предсказанная ошибка $\hat{s}$ в режиме IMA = 0,1 сильно переоценивает s (как и ожидалось), а при IMA = 2,3,4 слегка недооценивает; при IMA = 2 смещение минимально.

3. Остальные алгоритмы. Эффективность оценивания ошибки прогноза - проблема, можно сказать, вторичная, по сравнению с проблемой эффективности самого прогноза. Поэтому разумно срав-

нивать точность оценивания ошибки прогноза для различных режимов в рамках одного алгоритма.Между различными алгоритмами такое сравнение не всегда правомерно. Тем не менее нельзя не заметить,что именно при оценивании ошибки, в сравнении с другими алгоритмами,проявляет,наконец, себя грубость алгоритма ZL-CT. Хотя этот алгоритм во многих наборах данных дает очень хорошие прогнозы,но оценивает ошибку гораздо хуже,чем ZL-ПШ и ZL-НСПА. Можно было бы, конечно,вставить в него более изощренную оценку прогноза вроде кросс-проверки,но тогда он перестал бы быть самым быстрым алгоритмом семейства ZL.

§7. Псевдопробелы. d-редактирование

Обычное редактирование, к сожалению, не в состоянии ответить на вопрос, насколько хорошо работает тот или иной алгоритм заполнения пробелов в условиях действительного наличия какого-то процента пробелов. Может быть,при росте этого процента прогнозы алгоритма катастрофически ухудшаются? Естественно было бы попытаться "закрывать" в таблице не по одному элементу, как в обычном редактировании, а двойками, тройками, множествами $I_0 \subset I$, $|I_0| = d$.

Пусть, к примеру, нужно получить прогноз элемента z_{i_0}, $i_0 \in I$. Перебираем, скажем,тройки элементов и,"закрывая" их, превращаем в псевдопробелы.Находим прогноз элемента z_{i_0} :

$$A(Z, I\setminus\{i_0,i_1,i_2\}, i_0) = (\hat{z}_{i_0}[i_0,i_1,i_2], \dots).$$

А потом усредняем полученные прогнозы по всем сочетаниям (i_1,i_2):

$$\hat{z}^{(3)}_{i_0} = \Big(\sum_{(i_1,i_2)} \hat{z}_{i_0}[i_0,i_1,i_2]\Big)/C^2_{m-1},\ i_0 \neq i_1 < i_2.$$

В общем случае прогноз элемента z_{i_0} посредством d-*редактирования* определяется так:

$$\hat{z}_{i_0}^{(d)} = \Big(\sum_{I_0 \subset I} \hat{z}_{i_0}[I_0]\Big)/C_{mn-1}^{d-1}, \quad i_0 \in I_0, \ |I_0| = d,$$

где $A(Z, I \setminus I_0, i_0) = (\hat{z}_{i_0}[I_0], \hat{s}_{i_0}[I_0], s_{i_0}[I_0])$. Далее зависимость от множества элементов, объявленных псевдопробелами, будет подразумеваться неявно, а квадратные скобки будем опускать: $A(Z, I \setminus I_0, i_0) = (\hat{z}_{i_0}, \hat{s}_{i_0}, s_{i_0})$.

Описанный метод напоминает групповой метод "складного ножа" (delete-d-jackknife) [9, с.54], однако последний требует удалять не элементы таблицы, а целые строки, поскольку именно строки в нашем случае соответствуют элементам выборки.

Теперь рассмотрим d-редактирование множеств элементов. Обычное редактирование можно назвать 1-редактированием. В рамках проблемы сравнения алгоритмов представляет особый интерес режим редактирования самого множества псевдопробелов I_0:

$$A(Z, I \setminus I_0, I_0) = (\hat{Z}(I_0), \hat{S}(I_0), S(I_0)).$$

Во втором эксперименте d-редактирование активно применяется в целях сравнения алгоритмов и их режимов по точности прогнозирования (табл.7) и эффективности оценивания ошибки прогноза (табл.8) в условиях наличия заданного процента p пробелов.

Схема эксперимента состоит в следующем. Вместо трудоемких переборов используется метод Монте-Карло. Пусть задана таблица $Z(m \times n)$, p - процент удаления и k_0 - количество тактов моделирования.

На j-м такте генерируется очередное множество координат псевдопробелов $I_j = (i_1^{(j)}, \ldots, i_d^{(j)})$, $|I_j| = d$, $d = [pmn/100]$, $i_k^{(j)} = [\,]mv_1[\,,]nv_2[\,]$, где v_1, v_2 - два очередных случайных числа из $(0,1]$. Далее множество I_j редактирует алгоритм ZL: $A(Z, I \setminus I_j, I_j) = (\hat{Z}(I_j), \hat{S}(I_j), S(I_j))$,

Таблица 7

Точность прогноза при d-редактировании набора данных №1 (p = 10%). В скобках - значение параметра IMA

Алгоритм (Режим)	Фактическая квадратическая ошибка прогноза						
	Ql	Me	Qu	De	Int	$\bar{x}$	Sd
ПШ	8.1	12.1	17.1	24.0	4.5	16.4	16.2
НСПА(0,2)	11.8	13.9	19.3	29.7	3.7	17.5	11.2
НСПА(1,3)	11.0	15.3	21.9	31.5	5.4	18.9	12.6
НСПА(4)	10.9	14.4	19.7	27.4	4.4	17.2	10.2
СТ	8.8	12.7	17.5	31.4	4.3	17.2	14.6
НПШ(2)	8.7	12.0	16.9	26.5	4.1	15.7	11.8
НСТ(2)	9.8	13.5	18.4	24.9	4.3	15.8	10.8
НСПА+ПШ(2)	8.5	12.6	17.9	31.3	4.7	16.6	13.1
НСПА+СТ(2)	9.7	13.3	17.7	24.6	4.0	16.0	11.9
ЛСР	19.8	26.3	45.9	60.8	13.1	34.2	22.0

Таблица 8

Эффективность оценивания ошибки при d-редактировании набора данных № 1

Алгоритм (Режим)	Модуль разности ошибок прогноза					Разность ошибок	
	Ql	Me	Qu	De	Int	Me	Int
ПШ(0)	5.0	8.5	13.6	16.6	4.3	8.5	4.3
ПШ(1)	5.1	9.4	13.2	20.0	4.1	9.4	4.1
ПШ(2)	1.5	3.7	9.9	13.2	4.2	2.6	3.4
ПШ(3)	2.4	4.7	5.8	12.7	1.7	3.7	2.6
НСПА(0)	5.9	10.8	18.6	25.6	6.4	-10.0	6.9
НСПА(1)	3.2	5.8	10.6	15.5	3.7	5.6	3.8
НСПА(2)	2.2	4.5	8.5	13.9	3.2	4.5	3.2
НСПА(3)	6.5	10.1	15.5	25.4	4.5	-9.4	6.5
НСПА(4)	4.9	7.5	11.1	16.5	3.1	7.5	3.1
СТ	5.4	7.5	13.1	26.8	3.8	7.5	3.8

После завершения моделирования и редактирования с результатами поступаем так же, как и в предыдущем эксперименте, за исключением того, что вместо массивов $\{s_i\}, \{e_i\}, \{|e_i|\}$, $i \in I$, характеристики будут вычисляться над массивами $\{s_j^*\}, \{e_j^*\}, \{|e_j^*|\}$, $j = 1, \ldots, k_0$, где $s_j^* = \underset{i \in I_j}{\text{med}}\, s_i[I_j])$, а $e_j^* = \underset{i \in I_j}{\text{med}}\, (e_i[I_j])$, med - знак медианы. Почему мы взяли медиану, а не среднее ошибок в каждом множестве I_j? Среднее очень чувствительно к выбросам, а при не слишком малом p в каждую группу I_j наверняка попадет хотя бы один выброс, и каждым алгоритмом будет сделан хотя бы один очень плохой прогноз, так что в массиве $\{s_j^*\}$ почти все элементы испытают влияния выбросов. Это смазывает сравнительные эффективности алгоритмов, так как если где и проявляются четко различия алгоритмов в эффективности оценивания ошибки, то никак не на выбросах.

Результаты d-редактирования, примененного к набору данных №1 при $k_0 = 40$ и $p = 10\%$, представлены в табл.7 и 8. Если сравнить их с результатами обычного редактирования того же набора (табл.3 и 4), то можно подумать, что введение пробелов в таблицу приводит к резкому ухудшению плохих прогнозов (если судить по Qu, De, Int, $\bar{x}$, Sd). Однако на самом деле должно происходить обратное, просто на результатах отразилось сглаживающее влияние медианы. Эти таблицы нельзя сравнивать непосредственно по значениям характеристик, но можно сравнивать по рангам алгоритмов в иерархии каждой таблицы.

Если в табл.3 лучше всех предсказывал алгоритм ZL-СТ и связанные с ним алгоритмы типа ZL-НСПА+СТ, то теперь под воздействием пробелов он уступил первенство алгоритму ZL-ПШ и связанным с ним смешанным и составным алгоритмам. Смешанные алгоритмы работают здесь лучше, чем ZL-НСПА. В рамках алгоритма ZL-НСПА при 1-редактировании режим IMA = 4 был наилучшим, а при

d-редактировании режим IMA = 2 отвоевал у него "средние" характеристики (Me, Qu, Int), подтверждая все сказанное о нем в §6, п.5. Почему алгоритм ZL-НСПА при d-редактировании не смог дать более точных прогнозов, чем параметрические алгоритмы, ведь ранее был сделан вывод, что пробелы более неблагоприятно влияют на параметрические алгоритмы? Дело, видимо, в том, что набор данных №1 очень неоднороден, а идеальный случай для алгоритма ZL-НСПА - однородные данные с нелинейными зависимостями. Непараметрический прогноз есть взвешенное среднее, и поэтому к экстраполяции он совершенно не способен.

Перейдем к оцениванию ошибки. В ZL-ПШ по сравнению с 1-редактированием почти ничего не изменилось, режим ISHOS = 3 по-прежнему лидирует, правда, режим ISHOS = 2 теперь имеет меньшую медиану (Me). Зато в алгоритме ZL-НСПА полностью победил режим IMA = 2, хотя при 1-редактировании лучше работали IMA = = 4 и IMA = 0. Ошибку свою ZL-НСПА оценивает в целом лучше, чем ZL-СТ, но хуже, чем ZL-ПШ.

§8. Усечение ошибки. Отказ от прогноза

Если нашей целью является заполнение пробелов, а не редактирование комплектных элементов, то у нас нет другого способа судить о фактической ошибке прогноза ε, кроме как по предсказанной ошибке $\hat{\varepsilon}$. Как же бороться с плохими прогнозами? Если отказаться от прогноза, когда $\hat{\varepsilon}$ велика, то при условии, что $\hat{\varepsilon}$ достаточно эффективно оценивает ε, возможно, и фактическая ошибка ε редко будет большой.

В алгоритме заполнения пробелов ZET [13, с. 59] с этой целью интенсивно используются пороги, задаваемые пользователем. А именно: если $\hat{\varepsilon}_i > \varepsilon_0$, то $\hat{z}_i$ = PROB, ε_0 - порог. Но так ли легко пользователю задать порог, даже если ε - относительная ошибка, как это определено в ZET? Если даже пользователь точно знает, какая величина ошибки является приемлемой

для него, он может своим порогом заставить алгоритм отказаться от всех прогнозов (отсечь все) или, наоборот, ничего не отсечь, что вряд ли прибавит ему информации о данных. Кроме того, некоторые прогнозы с приемлемой предсказанной ошибкой, тем не менее, статистически могут являться выбросами, и вряд ли фактическая ошибка в таком случае будет приемлемой, разумнее было бы отсечь такие прогнозы.

Можно воспользоваться какой-нибудь устойчивой статистикой из многочисленных исследований, касающихся выбросов. Например, взять порог, предложенный Тьюки [14, с.61]:

$$s_0 = \underset{i\in I}{Qu}\,(\hat{s}_i) + 3\,\underset{i\in I}{Int}(\hat{s}_i)\ .$$

Но можно пожертвовать, скажем, p% наихудших прогнозов, чтобы остальные прогнозы в среднем имели большую точность. Следующий эксперимент как раз и посвящен проверке влияния p% усечения прогнозов на среднюю точность остальных и на эффективность оценивания ошибки. В первую очередь, это проверка того, насколько хорошо $\hat{s}$ оценивает большие значения s, т.е. способна ли в принципе $\hat{s}$ выявлять плохие прогнозы. Если она не в состоянии это делать, улучшения прогнозов не должно происходить. Кроме того, весьма вероятно, что в области своих малых значений $\hat{s}$ лучше оценивает s, чем в области больших, причем для разных алгоритмов это может проявляться в разной степени.

В эксперименте используются 10- и 25% усечения предсказанной ошибки $\hat{s}$, т.е. порог, равный верхней децили и квартили $\hat{s}$ соответственно: $s_0 = \underset{i\in I}{De}\,(\hat{s}_i)$, $s_0 = \underset{i\in I}{Qu}\,(\hat{s}_i)$. Пороги вычисляются при первом прогоне алгоритма (без усечения). В табл. 9 даны результаты применения алгоритма с усечением к набору данных №2. Алгоритмы в таблице сравниваются по точности прогноза.

Таблица 9

Редактирование с 10- и 25% усечением набора данных №2

Алгоритм (Режим)	Фактическая квадратическая ошибка прогноза						
	Ql	Me	Qu	De	Int	$\bar{x}$	Sd
10%							
ПШ(2)	11.1	25.4	50.7	96.4	19.8	39.1	40.5
ПШ(3)	11.1	25.3	50.4	95.8	19.6	38.0	38.8
НСПА(2)	16.8	32.5	55.4	89.3	19.3	39.7	31.9
НСПА(4)	17.1	29.0	54.1	87.9	18.5	40.3	33.4
СТ	12.5	26.7	49.6	105.9	18.5	41.5	43.0
25%							
ПШ(2)	9.8	23.5	40.5	68.5	15.4	32.7	36.3
ПШ(3)	9.8	23.1	41.7	76.9	15.9	32.2	32.2
НСПА(2)	16.1	27.1	45.3	77.6	14.6	36.5	30.8
НСПА(4)	18.1	29.7	52.4	87.9	17.2	40.0	32.9
СТ	12.1	24.3	47.0	108.6	17.5	40.0	44.2

Сравнивая результаты табл. 9 с результатами 1-редактирования этого же набора данных без усечения (табл. 5), можно прийти к следующим выводам.

1. Точность прогноза и эффективность оценивания фактической ошибки s при усечении предсказанной ошибки $\hat{s}$, несомненно, улучшились по всем характеристикам, что свидетельствует о способности оценки $\hat{s}$ к выявлению плохих прогнозов. Этот вывод относится ко всем алгоритмам и к обоим порогам (10- и 25%). Кроме того, это говорит о том, что при малых значениях $\hat{s}$ предсказанная ошибка лучше оценивает фактическую. Но у разных алгоритмов - в разной степени.

2. Если при редактировании без усечения базовые алгоритмы по точности прогноза распределялись так: 1) ZL-ПШ, 2) ZL-СТ, 3) ZL-НСПА, то при 25% усечении за счет того, что ZL-СТ хуже оценивает ошибку, чем ZL-НСПА, последнему удалось улучшить по сравнению с ZL-СТ характеристики Int и Qu и выйти на второе место.

3. В рамках алгоритма ZL-ПШ при отсутствии усечения прогноз во всех режимах один и тот же. Однако для каждого режима

усекается своя предсказанная ошибка, поэтому при p% усечении и прогнозы для различных значений ISHOS отличаются друг от друга. При 10% усечении по точности прогноза впереди режим ISHOS = 3, но уже для 25% усечения режим ISHOS = 2, наконец, использовал то, что при отсутствии усечения $\hat{s}$ лучше оценивает s именно для ISHOS = 2, и отвоевал себе несколько квантилей - Qu, De. Кстати, при 25% усечении $\hat{s}$ лучше оценивает s уже не для ISHOS = 2, а для ISHOS = 3.

4. В рамках алгоритма ZL-НСПА результаты очень хорошо иллюстрируют влияние усечения на точность прогноза, а также важность поиска хороших оценок для ошибки. При 1-редактировании без усечения режим IMA = 4 почти по всем характеристикам (кроме Ql, $\bar{x}$, Sd) предсказывал лучше, чем IMA = 2. Но при этом ошибку лучше оценивал режим IMA = 2 (по всем характеристикам, кроме Ql). При 10% усечении эти обстоятельства не вызвали никаких последствий, а при 25% усечении режим IMA = 2 по всем характеристикам имел более точные прогнозы, чем IMA = 4.

§9. Итерационная обработка пробелов

В алгоритме ZET [13, с.60] имеется режим итерационного заполнения пробелов, но он мало используется, так как эксперименты не обнаружили заметного улучшения прогнозов при использовании итераций. В алгоритмах семейства ZL также имеется режим обработки элементов с применением итераций. Проведены соответствующие эксперименты.

Обозначим исходную таблицу данных $(m \times n)$ через $Z_0 \equiv Z$. Если итерации не используются (количество итераций равно 1), то в общем случае схема алгоритма ZL задается формулой:

$$A(Z_0, I_p, I_0) = (\hat{Z}_0(I_0), \hat{s}_0(I_0), s_0(I_0)).$$

Определим элементы таблицы данных Z_k $(m \times n)$ на k-й итерации через элементы таблицы Z_{k-1} на $(k-1)$-й итерации.

Первый способ построения:

$$Z_k:\quad z_i^{(k)} = \begin{cases} \hat{z}_i^{(k-1)}, & \text{если } i \in I_0 \text{ и } \hat{z}_i^{(k-1)} \neq \text{PROB}; \\ z_i^{(k-1)} & \text{– в противном случае.} \end{cases}$$

Второй способ:

$$Z_k:\quad z_i^{(k)} = \begin{cases} \hat{z}_i^{(k-1)}, & \text{если } i \in I_0; \\ z_i^{(k-1)}, & \text{если } i \notin I_0. \end{cases}$$

Во втором способе при отказе от прогноза элемента z_i на $(k-1)$-й итерации этот элемент в таблице данных k-й итерации заменяется пробелом, в первом способе - элемент сохраняет свое значение, которое он имел на $(k-1)$-й итерации.

Теперь можно определить сам итерационный процесс.Для этого надо немного изменить схему алгоритма. Оператор A' в отличие от A дает на выходе матрицы $(m \times n)$ - $Z_k, \hat{S}_k, S_k$, а не векторы, как A :

$$A'(Z_0, I_p, I_0) = (Z_1, \hat{S}_1, S_1),$$
$$A'(Z_1, I_p, I_0) = (Z_2, \hat{S}_2, S_2),$$
$$\cdots \qquad\qquad \cdots$$
$$A'(Z_{k-1}, I_p, I_0) = (Z_k, \hat{S}_k, S_k).$$

Итерации можно применять совместно с усечением предсказанной ошибки. Порог s_0, введенный для p% усечения на первой итерации, в дальнейшем действует и на всех остальных; хотя разумнее, быть может, на каждой итерации задавать свой порог $s_0^{(k)}$.

Эксперименты, проведенные в случае итеративного 1-редактирования всей таблицы $I_0 = I_p = I$, показали, что в целом итерации не улучшают прогноза. Итерационные процессы в эвристических алгоритмах, как правило, работают плохо. Сходимо -

сти итерационных процедур следует добиваться теоретическими средствами, нужны теоретические исследования. К примеру, итерационный EM-алгоритм [15], судя по сообщениям, успешно работает при заполнении пробелов, и некоторые свойства сходимости этого алгоритма удалось доказать.

Все же в одном случае удавалось регулярно получать улучшение прогноза, а именно когда применялся алгоритм ZL-НСПА с усечением предсказанной ошибки. В табл. 10 представлены фактические ошибки прогноза на пяти итерациях при 25% усечении для алгоритма ZL-НСПА и режима IMA = 2, которые применялись к набору данных №1. По техническим причинам здесь вычисляются фактические относительные ошибки $\delta_i^{(k)} = 100 \cdot |\hat{z}_i^{(k)} - z_i| / |z_i|\%$, где k - номер итерации, а не квадратические $s_i^{(k)}$. Если судить по квантилям $\delta^{(k)}$, то на третьей итерации наступило заметное улучшение прогнозов, после чего фактическая ошибка стала расти, но уровня первой итерации достигнуть не успела.

Таблица 10

Итерационная обработка элементов набора данных №1 с 25% усечением. Алгоритм ZL-НСПА, режим IMA = 2

Номер итерации	Фактическая относительная ошибка прогноза						
	Ql	Me	Qu	De	Int	$\bar{x}$	Sd
1	28.8	50.8	81.5	137.7	26.3	72.7	86.7
2	20.6	50.0	83.7	133.1	31.6	75.2	105.0
3	18.6	46.8	78.7	129.3	30.0	74.1	112.1
4	19.2	47.1	80.1	133.0	30.4	71.5	104.0
5	19.1	43.4	80.9	133.9	30.9	72.9	111.5

§10. Метод "снова и снова"

Некоторые авторы предлагают свои алгоритмы заполнения пробелов, но не могут найти удовлетворительной оценки для ошибки прогноза. Тогда они довольствуются туманными соображениями типа "если на комплектных элементах таблицы мы имеем среднюю ошибку, скажем $s_{ср}$, то и прогноз пробела должен иметь приблизительно ту же фактическую ошибку". Действительно,в регрессионном анализе именно так и вводится оценка ошибки: предсказанная ошибка для пробела равна средней фактической ошибке для элементов того столбца, в котором находится пробел, разве что усреднение квадратическое и коэффициент присутствует:

$$\hat{\Delta}^2_{(0)} = (m_1 - n_1 - 1)^{-1} \sum_{i=1}^{m_1} \Delta^2_{(i)},$$

где $\Delta_{(i)} = |y_i - \hat{y}_i|$ (см.алгоритм ZL-ПШ, режим ISHOS = 0).

Однако эту идею можно реализовать в более общем виде, так как зависимость между элементами столбцов таблицы может порождать зависимость между фактическими ошибками прогноза для этих элементов. Предлагается следующий метод оценивания ошибки прогноза (*метод "снова и снова"*), использующий зависимости между фактическими ошибками.

Пусть алгоритм A' (как и в §9, рассматриваем модификацию алгоритма A) не умеет оценивать ошибку прогноза: $A'(Z,I,I) = (\hat{Z}, S_0)$. Здесь Z $(m \times n)$ - таблица данных с пробелами, S_0 - $(m \times n)$-матрица фактических ошибок прогноза. Если z_i =PROB, то $s_i^{(0)}$ = PROB.

Применим алгоритм A' к матрице S_0:

$$1)\ A'(S_0, I, I) = (\hat{S}_0, S_1),$$

и получим $\hat{S}_0$ - матрицу оценок фактических ошибок для прогнозов элементов S_0, в том числе и для пробелов.

Далее эти оценки можно уточнять итерационно ("снова и снова"):

$$2)\ A'(S_1,I,I) = (\hat{S}_1,S_2);$$

$$\cdots$$

$$k)\ A'(S_{k-1},I,I) = (\hat{S}_{k-1},S_k).$$

В результате мы получим приближенный доверительный интервал для прогноза элемента z_i, $i \in I$:

$$z_i = \hat{z}_i \pm \hat{s}_i^{(0)} \cdot t_{(p,m_1)}.$$

А в случае итерационного уточнения:

$$0)\ z_i = \hat{z}_i \pm s_i^{(0)} \cdot t_{(p,m_1)};$$

$$1)\ s_i^{(0)} = \hat{s}_i^{(0)} \pm s_i^{(1)} \cdot t_{(p,m_1)};$$

$$\cdots$$

$$k)\ s_i^{(k-1)} = \hat{s}_i^{(k-1)} \pm s_i^{(k)} \cdot t_{(p,m_1)}.$$

Разумеется, совершенно не обязательно искать зависимости непосредственно между фактическими квадратическими ошибками s_i . Можно было бы подставить вместо $S_0,S_1,\ldots$ матрицы каких-нибудь других величин, из которых потом легко получаются s_i , например, матрицы $E_0,E_1,\ldots$, состоящие из остатков $\varphi_i = z_i - \hat{z}_i$, $i \in I$, или $\Delta_i = |\varphi_i|$; а в случае ZL-ПШ, следовало бы поискать зависимости между квадратами остатков $\Delta_i^2 = (z_i - \hat{z}_i)^2$ и т.д. Соответственно предсказывались $\hat{\varphi}_i$, $\hat{\Delta}_i$, $\hat{\Delta}_i^2$, из которых вычислялись бы предсказанные ошибки $\hat{s}_i$.

Были проведены эксперименты,в которых при обычном редактировании набора данных №1 квадратическая ошибка оценивалась по методу "снова и снова" (с одной итерацией). Вместо S_0 алгоритм применялся к матрице остатков E_0 :

$$A'(Z,I,I) = (\hat{Z},S_0);$$

$$A'(E_0,I,I) = (\hat{E}_0,S_1).$$

После преобразования $\hat{E}_0$ в $\hat{S}_0$ были сформированы массивы результатов $\{e_i\}$, $e_i = s_i^{(0)} - \hat{s}_i^{(0)}$ и $\{|e_i|\}$, $i \in I$, характеристики которых приведены в табл. 11. Эти результаты вместе с результатами табл. 3 позволяют сравнить оценки ошибки по методу "снова и снова" и оценки ошибки, введенные в §3-4.

Таблица 11

Метод "снова и снова". Набор данных №1

Алгоритм (Режим)	Модуль разности ошибок прогноза					Разность ошибок	
	Ql	Me	Qu	De	Int	Me	Int
ПШ	6.7	19.3	57.6	177.6	25.5	-0.1	18.3
НСПА(0,2)	3.5	11.6	48.7	183.0	22.6	6.0	19.9
НСПА(1,3)	4.4	11.6	43.1	170.0	19.4	5.7	17.0
НСПА(4)	4.1	11.0	50.1	174.5	23.0	6.4	18.8
СТ	7.6	16.5	67.1	224.8	29.7	-2.6	16.7
ЛСР	4.3	10.4	39.4	265.5	17.5	1.9	20.2

Очевидно, что метод "снова и снова" оценивает ошибку довольно грубо и, конечно, хуже, чем специфические оценки алгоритмов. Для алгоритма ZL-НСПА он оценивает ошибку эффективнее, чем для параметрических алгоритмов, поскольку последние выявляют линейные зависимости, а ошибки даже в случае линейной модели теоретически связаны нелинейно .

В целом, за неимением лучшего, можно использовать и этот метод.

Заключение

Результаты экспериментов и выводы из них свидетельствуют о важности поиска эффективных оценок для ошибки прогноза при заполнении пробелов и редактировании элементов. Были выяснены условия, при которых в том или ином алгоритме семейства ZL следует применять тот или иной режим оценивания. Особенно хорошо показали себя режимы ISHOS = 3 (ZL-ПШ) и IMA = 2 (ZL-НСПА).

Ряд примеров позволяет во многих случаях улучшать прогнозы и оценки ошибки. Рассмотренные алгоритмы - каждый в своем классе данных - проявляют себя неплохо. В целом,если нужны недвусмысленные и четкие рекомендации, то алгоритм ZL-ПШ, по-видимому, оптимально сочетает в себе экономичность и точность прогнозов по сравнению с другими алгоритмами семейства ZL и может быть рекомендован к применению в первую очередь.

Литература

1. ЗАГОРУЙКО Н.Г., УЛЬЯНОВ Г.В. Локальные методы запол - нения пробелов в эмпирических таблицах //Экспертные системы и распознавание образов. - Новосибирск. - 1988. - Вып. 126: Вычислительные системы. - С. 75-103.

2. ЗАГОРУЙКО Н.Г., УЛЬЯНОВ Г.В. Заполнение пробелов в 3-входовых таблицах данных типа "объект-свойство-время"//Там же. - С. 104-121.

3. ЛБОВ Г.С. Методы обработки разнотипных экспериментальных данных. - Новосибирск: Наука, 1981. - 160 с.

4. ХЬЮБЕР П. Робастность в статистике. - М.: Мир, 1984. - 303 с.

5. CHATTERJEE S., HADI A.S. Influential observations,high leverage points, and outliers in linear regression //Statist. Science. - 1986. - Vol. 1. - P. 379-416.

6. ДЕМИДЕНКО Е.З. Линейная и нелинейная регрессия. - М.: Финансы и статистика, 1982. - 300 с.

7. ALLEN D.M. Mean square error of prediction as a criterion of selecting variables //Technometrics. - 1971. - Vol.13. - P. 469-475.

8. STONE C.J. Cross-validatory choice and assesment of statistical predictions //J.Roy.Statist. Soc.B. - 1974.-Vol.36. - P.111-147.

9. ЭФРОН Б. Нетрадиционные методы многомерного статистического анализа. -М.: Финансы и статистика, 1988. - 262 с.

10. GEISSER S. The predictive sample reuse method with applications //J.Amer. Statist. Assoc. - 1975. - Vol.70.-P. 320-328.

11. RAO C.R. Prediction of future observations in growth curve models //Statist. Science. - 1987. - Vol. 2. -P.434-471.

12. Математическое обеспечение ЕС ЭВМ. - Минск,1978.-Вып. 15 (Ин-т математики АН БССР).

13. ЗАГОРУЙКО Н.Г., ЁЛКИНА В.Н., ЛБОВ Г.С. Алгоритмы обнаружения эмпирических закономерностей. - Новосибирск: Наука, 1985. - 108 с.

14. ТЬЮКИ Дж. Анализ результатов наблюдений: Разведочный анализ данных. - М.: Мир, 1981. - 693 с.

15. LITTL R.J., RUBIN D.B. Statistical analysis with mis - sing data. - N.J.: Wiley, 1987. - 278 p.

16. RUBIN D.B. Using multiple imputations to handle nonresponse in sample survays. - N.J.: Wiley, 1987. - 258 p.

Поступила в ред.-изд.отд.

12 июня 1989 года