МЕТОДЫ И СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА
Вычислительные системы
1992 год Выпуск 145

УДК 62-5:007:621.391

СИНХРОНИЗАЦИЯ В ЗАДАЧЕ ЗАПОЛНЕНИЯ ПРОПУЩЕННЫХ ЗНАЧЕНИЙ В ЭМПИРИЧЕСКИХ ТАБЛИЦАХ ТИПА "ОБ'ЕКТ-СВОЙСТВО-ВРЕМЯ"

Н.Г.Загоруйко, О.В.Кужелев

Проблеме анализа таблиц данных с пропусками уделяется все большее внимание со стороны разработчиков программного обеспечения. Положение, когда в их программах преобладают такие примитивные приемы, как отбрасывание всего наблюдения при наличии в нем хотя бы одного пропуска, заполнение пропущенного значения средним по столбцу-свойству, в последнее время начинает меняться. Недостатки упомянутых способов очевидны, это: потеря более или менее значительной части информации в первом случае и смещение оценок параметров распределения - во втором. Более обоснованным методом представляется заполнение пропусков с помощью регрессии, хотя и здесь имеет место уменьшение дисперсии отклика за счет условной по предиктору компоненты. Приблизительно такой подход реализуется в одном из известных алгоритмов - алгоритме ZET [1]. Важной особенностью ZET является то, что это - локально-параметрический метод: пропуск заполняется в локальной подматрице таблицы. В [2] продолжается линия развития этого метода для статических таблиц. В [3] эта же концепция распространяется уже на трехмерные таблицы (объект-свойство-время). Данное обстоятельство, а именно, появление времени и вместе с ним некоторых специфических особенностей в механизме заполнения, обсуждается в данной работе.

Зависимости в двумерной таблице "объект-свойство" являются пространственно-статическими в том смысле, что данные в ней сняты в один фиксированный момент времени. Объекты же рассматриваются как точки в многомерном по числу свойств признаковом пространстве, в котором определяют различного типа расстояния между объектами, например, эвклидовы расстояния. Эти расстояния тем больше, чем более "непохожи" друг на друга объекты. При появлении временной размерности двумерная таблица превращается в трехмерную (трехвходовую) и возникают более сложные зависимости, которые можно назвать уже пространственно-динамическими. Отличие здесь не просто количественное в виде добавления одной размерности к имеющимся двум. Теперь требуется, чтобы вся динамика объекта охватывалась в единое целое и рассматривалась, возможно, как некоторый процесс. Применение в этом случае статических примеров не может быть удовлетворительным, что наглядно демонстрирует следующий пример. На рис.1 изображены графики трех одномерных динамических объектов. Если рассматривать их как процессы, то видно, что 1-й и 2-й протекают сходным образом, или, последовательности их состояний повторяют друг друга, правда, со сдвигом во времени. Третий же объект ведет себя несколько иначе. Однако вычисление расстояний между ними по классической в статике схеме

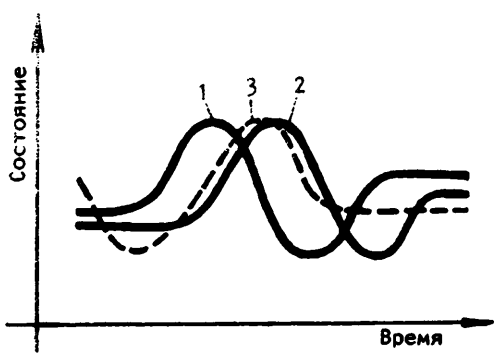

Рис. 1

$$\left(\sum_{k=1}^{l} |x_k^i - x_k^j|^r\right)^{1/r}$$

(где l - число точек времени, x_k^i - значение свойства i-го объекта в момент времени k)

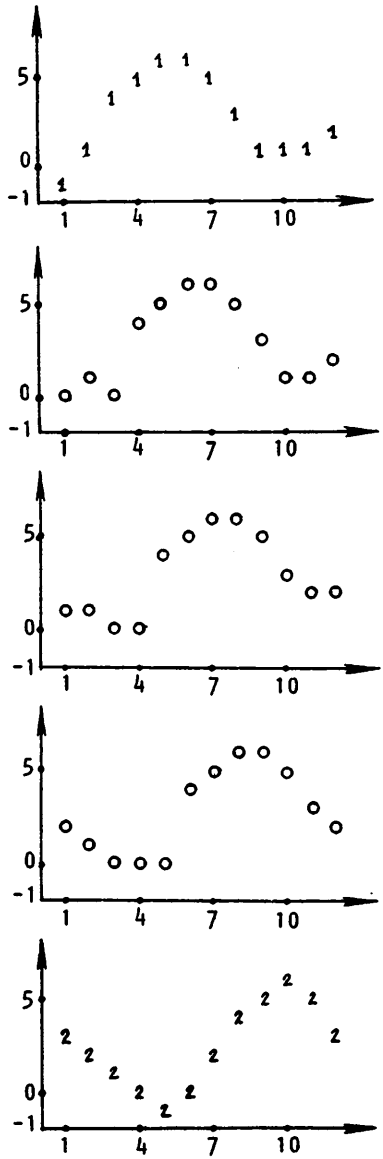

Рис.2

приведет к иным результатам: кривые 2 и 3 окажутся более "похожими" друг на друга, чем кривые 1 и 2. Следовательно, в трехвходовых таблицах можно обнаружить качественно новые по сравнению с двухвходовыми типы зависимостей, и это надо учитывать.

В [4] сравнивается традиционный подход с методом, выполняющим синхронизацию применительно к классификации именно такого рода динамических объектов. Сравнение оказывается в пользу последнего метода, обозначенного как D-метод. Его преимущество объясняется тем, что при вычислении расстояния он автоматически ускоряет либо замедляет время, иначе говоря, стремится сжать либо растянуть формы графиков динамических объектов так, чтобы "подогнать" их друг к другу. (На рис.2 показан процесс поэтапной подгонки графика 1 к графику 2.) Это делается с целью синхронизировать последовательности состояний объектов, что позво-

ляет не пропустить сходство в поведении динамических объектов, если это сходство имеет место, несмотря на то, что процессы могут идти с разной скоростью. Аналогично в алгоритмах, заполняющих пропуски в таблицах "объект-свойство-время", следовало бы предусмотреть возможность устранения имеющегося временного несоответствия. Ни ZET, ни другие алгоритмы этого не делают.

В описываемом ниже алгоритме DPZ делается попытка включения процедуры синхронизации процессов в заполнение пропусков из трехвходовой таблицы. Итак, рассматривается таблица "объект-свойство-время" $T = \{t_{ijk}\}$, где t_{ijk} есть значение свойства $j = \overline{1,n}$ i-го объекта $(i = \overline{1,m})$ в момент времени $k = \overline{1,l}$. Таблица T, с одной стороны, состоит из двумерных таблиц одного из трех типов, например, типа "свойство-время", каждая из которых соответствует некоторому объекту. Можно сказать, такая таблица полностью определяет один динамический объект. С другой стороны, таблицу T составляют одномерные массивы: строки, когда фиксируются индексы i,k, столбцы – при фиксировании j,k, ряды - при фиксированных i и j. В DPZ центральное место отводится рядам таблицы T. Делается это из предположения, что из перечисленных одномерных составляющих - столбцов, строк и рядов - именно последние обладают динамикой, свойствами процессов.

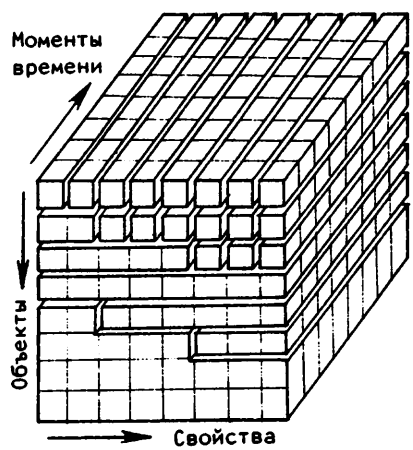

Рис. 3

Перед заполнением пробелов элементы таблицы нормируются к интервалу $[0,1]$. Нормиров-

ка делается отдельно для каждой из n двумерных таблиц-сечений типа "объект-время" - вертикальных продольных слоев куба (рис.3)

Пусть отсутствует значение $t_{i_0 j_0 k_0}$. В первую очередь отыскивается ряд, которому принадлежит данный пропуск. Этот ряд $\{t_{i_0 j_0 k_u}\}$ содержит $l_0 \le l-1$ элементов. Для него определяются ряды-аналоги, т.е. ряды таблицы T, нормализованные D-расстояния [4] до которых от ряда $\{t_{i_0 j_0 k_u}\}$ наименьшие. Допустим найдены p рядов-аналогов - $\{t_{i_1 j_1 k_u}\}$ ..., ..., $\{t_{i_p j_p k_u}\}$ с нормализованными расстояниями D_1,..., D_p. Число l_s элементов в $\{t_{i_s j_s k_u}\}$, где $s = \overline{1,p}$, меньше либо равно l из-за возможности наличия других пропусков. Берется пара рядов: тот, что с пропуском - $\{t_{i_0 j_0 k_u}\}$, и какой-нибудь, например, первый из аналогов - $\{t_{i_1 j_1 k_u}\}$. Оба они имеют графическое представление в виде последовательности точек на плоскости, где по оси абсцисс откладывается время. Соединенные отрезками, точки образуют ломаные кривые. Процедура синхронизации, основанная на динамическом программировании, деформирует эти кривые таким образом, что каждой точке одной кривой ставятся в соответствие некоторые точки другой (одна или несколько). Значит, каждому элементу из $\{t_{i_0 j_0 k_u}\}$ будут соответствовать один или более элементов из $\{t_{i_1 j_1 k_u}\}$. Называется данное соответствие, несмотря на неоднозначность, функцией деформации. Механизм синхронизации действует следующим образом. Во-первых, строится таблица $l_0 \times l_1$, в клетки которой вносятся значения $|t_{i_0 j_0 k_u}$ -

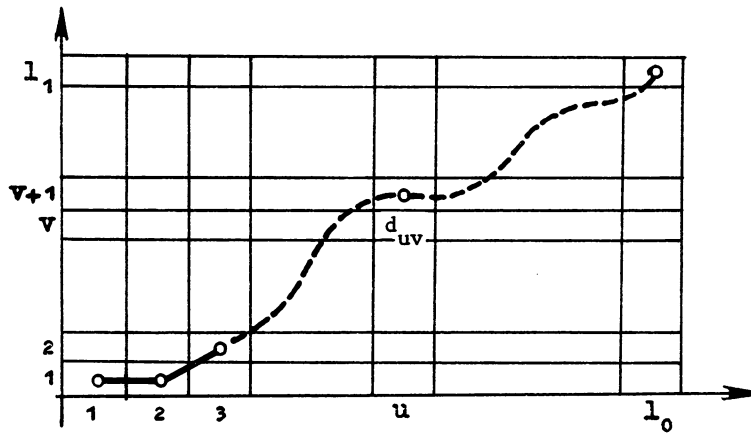

Рис. 4

$- t_{i_1 j_1 k_v}| = d_{uv}$, где $u = 1,\ldots,l_0$, $v = 1,\ldots, l_1$ (рис.4). Во-вторых, определяется функция деформации.

Искомая функция есть некоторый путь из левого нижнего угла таблицы в правый верхний и обладает свойствами непрерывности, а также монотонности, т.е. сектор направлений пути строго ограничен направлениями вправо и вверх.

Синхронизация достигается при условии прохождения функции деформации по клеткам с минимальными значениями d_{uv}, точнее, по пути, для которого Σd_{uv} минимальна. Избежать полного перебора всех возможных путей в таблице $l_0 \times l_1$ помогает динамическое программирование. Столбец за столбцом по рекуррентной формуле

$$dist_{uv} = d_{uv} + \min(dist_{u-1,v}, dist_{u-1,v-1} + d_{uv}, dist_{u,v-1})$$

вычисляется минимальное общее расстояние $dist_{l_0 l_1}$ и вместе с тем определяется оптимальный путь.

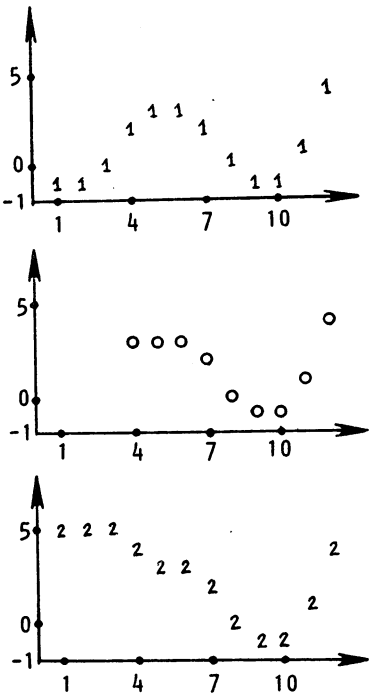

Рис. 5

Пока предполагалось, что начало пути находится в нижнем левом углу, а конец - в верхнем правом, т.е. $u_{нач}$, $v_{нач} = 1$, $u_{кон} = l_0$, $v_{кон} = l_1$. В DPZ существует возможность варьирования граничных условий. В этом случае синхронизируются уже не только ряды в целом, но и их усеченные участки. Ведь по - чему бы не предположить, что в процессах, в общем протекающих по-разному, присутствуют очень похожие фрагменты. Рис. 5 демонстрирует способность DPZ выявлять их. Для рассматриваемой пары рядов находится столько функций деформации, сколько подбирается граничных условий. Из них выбирается та, которой отвечают такие значения $u_{нач}$, $v_{нач}$, $u_{кон}$ и $v_{кон}$, что достигается минимум выражения

$$\frac{dist(u_{нач}, v_{нач}, u_{кон}, v_{кон})}{(u_{кон} + v_{кон} - u_{нач} - v_{нач} + 2)}.$$

Минимум этот называется нормализованным D-расстоянием. Для значений начального и конечного u, индексирующего элементы ряда $\{t_{i_0 j_0 k_u}\}$, имеется одно ограничение, от которого сво-

боден индекс v. Если выкинутый из $\{t_{i_0 j_0 k_u}\}_0$ пропуск $t_{i_0 j_0 k_0}$ поставить на свое место, то он имел бы двух либо, если его место крайнее, одного ближайшего соседа. Граничные условия при варьировании отсекают "хвосты" рядов с обеих сторон, образуя таким образом фрагменты ряда $\{t_{i_0 j_0 k_u}\}_0$. Ограничение на $u_{нач}$ и $u_{кон}$ состоит в том, чтобы, по крайней мере, один из ближайших соседей пропуска $t_{i_0 j_0 k_0}$ всегда присутствовал во фрагменте $\{t_{i_0 j_0 k_u}\}_0$.

Итак, функция деформации установила для элементов из $\{t_{i_0 j_0 k_u}\}_0$ соответствующие элементы, или образы, из первого ряда-аналога. Правда, для пропуска $t_{i_0 j_0 k_0}$ функция деформации не устанавливает образ автоматически, так как не определена на нем в силу того, что его значение неизвестно. Алгоритм DPZ определяет образ для пропуска как среднее арифметическое образов его ближайших соседей.

Все то же самое повторяется с использованием одного за другим остальных аналогов. В результате получается p образов для пропуска $t_{i_0 j_0 k_0}$: $t_1, \ldots, t_p$. Если принять, не нарушая общности, что нормализованные D-расстояния удовлетворяют неравенствам $D_1 \le D_2 \le \ldots \le D_p$, то формула предсказания значения $t_{i_0 j_0 k_0}$ будет иметь следующий вид:

$$\frac{\min(\Pi,\alpha)\left[\sum_{s=1}^{p-1} t_s \left(\frac{D_p - D_s}{D_p - D_1}\right)^f \Big/ \sum_{s=1}^{p-1}\left(\frac{D_p - D_s}{D_p - D_1}\right)^f\right] + \max(0,\Pi-\alpha)\,t_{ZET}}{\Pi},$$

где t_{ZET} - прогноз алгоритма ZET. α - процент ожидаемой ошибки t_{ZET} (вычисляется в алгоритме ZET), Π - порог, из-

меряемый как и α, в процентах. Видно, что будет отсеиваться влияние t_{ZET}, когда процент его ожидаемой ошибки превысит этот порог: $\alpha > \Pi$; напрашивающееся значение - $\Pi \sim 5\text{-}10\%$. Однако эмпирически, в ходе нижеописываемых вычислительных экспериментов, выяснилось, что лучше, когда $\Pi = \Pi(\alpha)$. Близкий к оптимальному вид функции оказался $\Pi = 30/\sqrt{2^{\alpha}}$. Показатель степени f предоставляется задавать пользователю в зависимости от желания еще увеличить (при большем f) либо уменьшить (при меньшем f) вклад в прогноз более похожих аналогов по сравнению с менее похожими.

Таковы ключевые моменты алгоритма DPZ. Было проведено сравнительное испытание его с методом ZET на нескольких вариантах таблицы T типа "объект-свойство-время". Для каждого варианта проводилась серия испытаний. Данные строились искусственно, на основе элементарных функций, полиномиального либо тригонометрического типа. Следовательно, первоначально сознательно устанавливались достаточно четкие функциональные зависимости внутри T (рис.6). Распределение пропусков в таблице близко к равномерному, насколько позволил датчик псевдослучайных чисел. Практически был выполнен принцип ОПС [5] - данные отсутствуют полностью случайно (MCAR - missing completely at random). Доля пропусков обычно не превышала 5%. Оба метода ведут себя в этом случае удовлетворительно, но ZET все же предпочтительней, так как он быстрее работает.

В следующих сериях испытаний в таблицу данных вводились искажения в те зависимости, которые направлены вдоль оси времени. Виды искажений - сжатие-растяжение и сдвиг. И тот, и другой виды искажений - линейные, т.е. постоянны в пределах каждого из рядов, однако от ряда к ряду степень и направление (положительное либо отрицательное) их меняется непредсказуемо, хаотично, также по датчику псевдослучайных чисел.

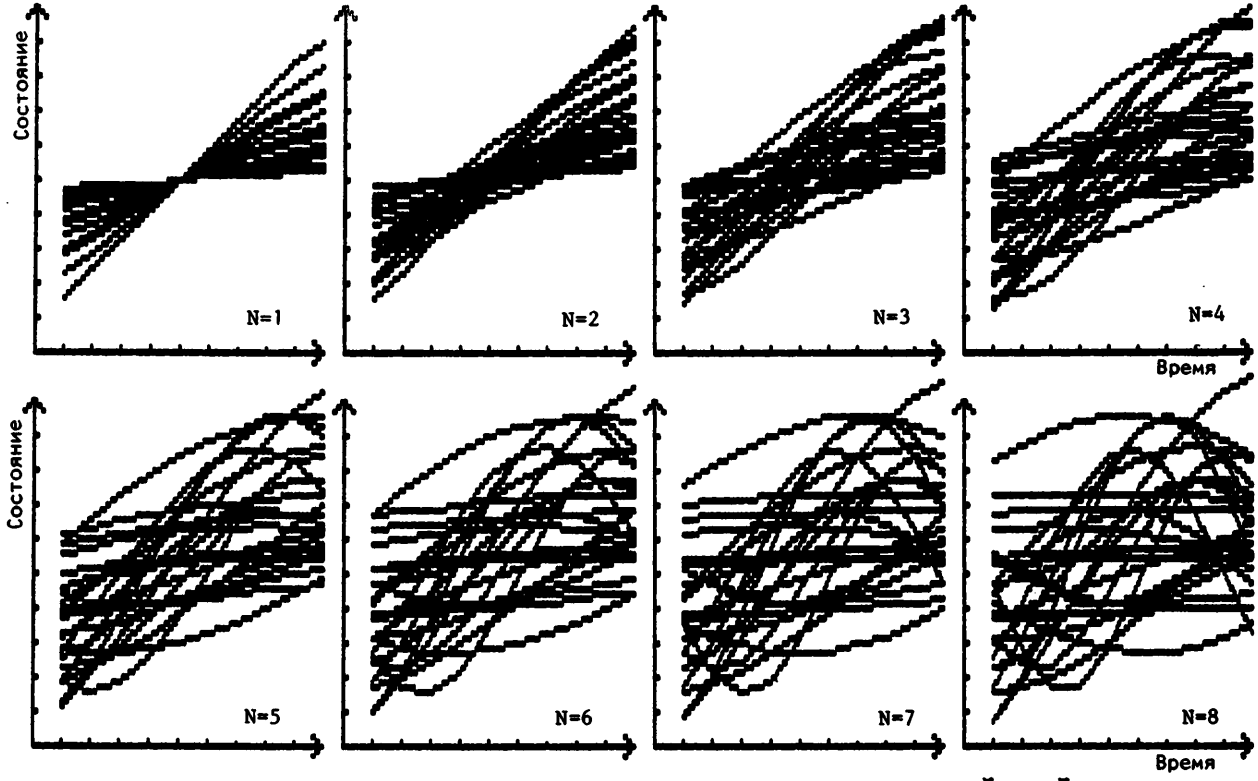

Рис.6. Графическое изображение группы объектов при различных значениях ξ_1 и ξ_2

Искажения усиливались от испытания к испытанию. Рост ошибок в прогнозах был отмечен в обоих методах, только у ZET он шел более прогрессивно. Число p рядов-аналогов в экспериментах обычно не превышало 4-5. Показатель степени f задавался значениями 1,2,3. Применялось фрагментирование рядов, осуществлявшееся посредством варьирования граничных условий функции деформации. Оно оказалось небесполезным, несколько улучшило прогнозы. Недостаток DPZ - затрачиваемое машинное время, которое приблизительно на порядок больше, чем у самых быстродействующих версий ZET. Для ЭВМ, выполняющей миллион операций в секунду, обсчет "нормальных" по объему данных занимает несколько десятков секунд. Одна конкретная, но типичная серия испытаний, а также количественное описание результатов этой серии приводятся ниже.

Значения m, n, l (число объектов, свойств, моментов времени): $m = 10$, $n = 5$, $l = 10$. Первый столбец таблицы ($k = j = 1$, $i = 1,\ldots, m$) составляют целые числа 2,3,4,5, 4, 3,4,3,3,3. Остальные клетки T формировались так:

$$t_{ij1} = t_{i11}\,2j/n, \qquad i = 1,\ldots,m,\ j = 2,\ldots,n,$$

$$t_{ijk} = t_{ij1}\sin\left(\frac{\pi}{2}\left(\frac{k}{l} - \frac{1}{2}\right)\xi_1 + \xi_2\right),$$
$$i = 1,\ldots,m/2,\ j = 1,\ldots,n,\ k = 2,\ldots,l,1,$$

$$t_{ijk} = t_{ij1}\cos\left(\frac{\pi}{2}\left(\frac{k}{l} - \frac{1}{2}\right)\xi_1 + \xi_2\right),$$
$$i = m/2+1,\ldots,m,\ j = 1,\ldots,n,$$
$$k = 2,\ldots,l,1.$$

Случайные величины ξ_1 и ξ_2, служащие для внесения искажений сжатия-растяжения (ξ_1) и сдвига (ξ_2), постоянны для элементов, принадлежащих некоторому ряду, и различны для элементов из разных рядов. Интервалы разброса ξ_1 и ξ_2,

а вместе с ними и искажения возрастают с каждым испытанием серии, но для первого испытания $\underline{\xi}_1 = 1$, $\underline{\xi}_2 = 0$. Зависимость нижних $(\underline{\xi}_1, \underline{\xi}_2)$ и верхних $(\overline{\xi}_1, \overline{\xi}_2)$ границ интервалов значений этих случайных величин от номера испытания N определяется следующими равенствами:

$$\underline{\xi}_1 = 1 - 0{,}1\,(N-1), \qquad \overline{\xi}_1 = 1 + 0{,}4\,(N-1),$$

$$\underline{\xi}_2 = -\pi \cdot 0{,}09\,(N-1) \qquad \overline{\xi}_2 = \pi \cdot 0{,}09\,(N-1).$$

Графические изображения той группы рядов, что порождены синусной функцией, даны на рис.6. Хорошо видно, как с возрастанием номера испытания N под влиянием искажений трансформируются правильные формы графиков. Число пропусков в таблице равно 15 (3% от всего объема данных). Распределение пропусков - ОПС. Количество рядов-аналогов $p = 4$.

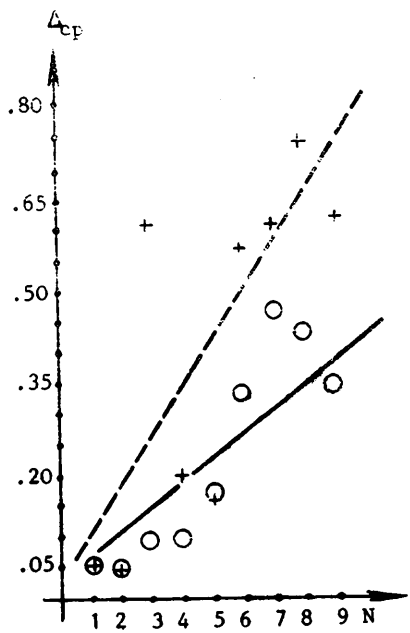

Рис.7

Полученные результаты удобно представляются на графике (рис.7), где вдоль горизонтальной оси отложены номера испытаний, а по вертикальной - среднее значение ошибок прогноза. Под ошибкой прогноза подразумевается модуль разности между прогнозом и действительным значением пропуска, которое заранее известно из-за искусственного характера данных. Крестики и пунктирная линия на рис.7 относятся к результатам алгоритма ZET, кру-

жочки и сплошная линия - к результатам алгоритма DPZ. Более крутая линия означает более быстрый рост ошибок. Неплохо работая в первых двух испытаниях, ZET "спотыкается" уже на третьем, и во всех следующих, за исключением пятого, значительно усту - пает конкуренту. Последний, напротив, демонстрируя стабиль - ность в течение первых четырех испытаний, затем постепенно сда- ет позиции, но не столь стремительно, как это делает ZET.

Проведенные вычислительные эксперименты позволяют сделать вывод о целесообразности использования процедуры синхронизации при обработке данных из таблиц типа "объект-свойство-время". Без этого характерные в некоторых приложениях временные сдвиги неизбежно влекут за собой неточности в работе классификацион - ных либо прогностических методов. Как показывает последний при- мер, на определенном этапе происходит резкое превращение оши - бок в суперошибки. Таким образом, обращение с динамическими объектами нуждается в некоторых специфических приемах, одним из которых является синхронизация.

Литература

1. ЁЛКИНА В.Н., ЗАГОРУЙКО Н.Г., НОВОСЕЛОВ Ю.А. Математические методы агроинформатики.- Новосибирск, 1987.-С.85-104.

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

3. Их же. Заполнение пробелов в 3-входовых таблицах данных типа "объект-свойство-время" //Там же.- С.104-121.

4. ЗАГОРУЙКО Н.Г., КУЖЕЛЕВ О.В. Синхронизация многомерных динамических процессов // Анализ данных и знаний в экспертных системах.-Новосибирск, 1990.- Вып.134: Вычислительные систе - мы.- С.85-95.

5. ЛИТТЛ Р.Дж., РУБИН Д.Б. Статистический анализ данных с пропусками/ Пер. А.М.Никифорова.- М.: Финансы и статистика, 1991.

6. САМБУР М.Р., СМИТ А.Р. Построение гипотез о словах и проверка слов для распознавания речи// Методы автоматического распознавания речи. Кн.1.- М.,1983.

Поступила в ред.-изд.отд.

6 июля 1992 года