УДК 004.93'14:004.853

РЕШЕНИЕ ЗАДАЧИ КЛАСТЕРИЗАЦИИ МЕТОДОМ КОНКУРЕНТНОГО ОБУЧЕНИЯ ПРИ НЕПОЛНЫХ СТАТИСТИЧЕСКИХ ДАННЫХ *

А.С. Ефимов

Нижегородский госуниверситет им. Н.И. Лобачевского

anton.yefimov@mail.ru

Поступила в редакцию 15.09.2009

Представлен обзор современных методов восстановления пропусков в неполных статистических данных. Предложен способ совмещения процедуры кластеризации статистических данных и обработки пропущенных в них значений на основе модификации алгоритма конкурентного обучения сети Кохонена. Показана эффективность способа при решении задачи определения индивидуальных доз радиоактивного йода-131 при лечении больных диффузным токсическим зобом.

Ключевые слова: кластеризация, конкурентное обучение, пропуски в статистических данных.

Введение

В настоящее время гибридные системы искусственного интеллекта стали активно применяться при решении широкого круга задач классификации и прогнозирования. В основе одной из реализаций систем данного класса лежит нечеткая экспертная система, база знаний которой генерируется в процессе структурной и параметрической идентификации на основе имеющихся в наличии статистических данных. Идентификация системы осуществляется в процессе проведения кластеризации доступных в виде баз данных статистических данных с последующим отображением структуры кластеров в структуру нечеткой базы знаний. Однако в большинстве случаев имеющиеся базы данных имеют значительное количество пропусков в таблицах. Объективными причинами этого являются поломки оборудования при измерении тех или иных характеристик, потеря ретроспективной информации, ограничение доступа к информации и другие. Субъективные причины обусловлены человеческим фактором при накоплении и обработке информации. Таким образом, необходимым условием построения гибридных систем данного класса является привлечение того или иного способа обработки или предварительного восстановления пропусков в статистических данных при проведении их кластеризации.

1. Обзор современных методов восстановления пропусков в данных

В настоящее время разработано множество методов восстановления пропусков в таблицах баз данных [1–4]. Перечислим наиболее распространенные методы данного класса с указанием их основных особенностей:

1) *Исключение строк с наличием пропусков.* Данный метод легко реализуем, но необходимым условием его применения является следование данных требованию MCAR (missing completely at random), т.е. пропуски в данных по переменным должны быть полностью случайными [1]. Кроме того, он обычно применяется лишь при незначительном количестве пропусков в таблице, иначе полученная на выходе таблица данных становится непредставительной. Главный недостаток такого подхода обусловлен потерей информации при исключении неполных данных.

2) *Заполнение пропусков средними по столбцу значениями.* Данный метод также легко реализуем, но его применение имеет смысл только в случае удовлетворения данных условию MAR (missing at random), т.е. когда пропуски в данных по переменным являются случайными и сам механизм пропусков несущественен [1]. К недостаткам метода относят вносимые искажения в распределения данных, уменьшение дисперсии.

3) *Метод ближайших соседей.* В основе метода лежит механизм поиска строк таблицы, которые по определенному критерию являются ближайшими к строке с пропусками. Для заполнения пропуска значения данной переменной (в фиксированном столбце) у соседних

* Статья рекомендована к печати программным комитетом Международной научной конференции «Параллельные вычислительные технологии 2009» (http://agora.guru.ru/pavt).

строк усредняются с определенными весовыми коэффициентами, обратно пропорциональными расстоянию к строке с пропуском. При большом количестве пропусков данный метод также практически неприменим, поскольку базируется на существовании связей между строками в таблице.

4) *Регрессионный анализ.* Из условий применения данного метода можно выделить требование о следовании данных условию MAR (хотя для частных случаев возможно применение более слабых требований) и требования, относящиеся к выполнению предпосылок регрессионного анализа. Недостатки метода очевидны: качество восстановления пропусков напрямую зависит от успешного выбора взятой за основу регрессионной модели.

5) *Метод сплайн-интерполяции.* Для успешного применения необходимо, чтобы данные следовали условию MAR. Недостатки метода следуют из самой его идеи. Например, в случае восстановления группы пропусков, следующих подряд друг за другом, результат аппроксимации сплайном данной группы не всегда может дать оценки, приближающиеся с достаточной точностью к значениям, которые могли бы быть на месте пропусков.

6) *Метод максимальной правдоподобности и EM-алгоритм.* Метод требует проверки гипотез о распределении значений переменных. Применение осложняется при большом количестве пропущенных значений переменной. Особенность данного метода состоит в построении модели порождения пропусков с последующим получением выводов на основании функции правдоподобия, построенной при условии справедливости данной модели, с оцениванием параметров методами типа максимального правдоподобия. Отметим, что для данных методов возможно построение моделей, учитывающих конкретную специфику области, и, как следствие, возможна постановка более слабых условий к данным (слабее MAR).

7) *Алгоритмы ZET и ZetBraid.* По сути, алгоритм ZET является детально проработанной и апробированной технологией верификации экспериментальных данных, основанной на гипотезе их избыточности. Главная идея алгоритма ZET заключается в подборе «компетентной матрицы», используя данные из нее находят параметры зависимости, которая применяется для прогнозирования пропущенного значения. Субъективизм определения размерности «компетентной матрицы» приводит к учету неинформативных и шумовых факторов и смещению оценки неизвестного значения. Основное отличие алгоритма ZetBraid состоит в определении оптимального размера «компетентной матрицы». Данные алгоритмы хорошо показали себя, но статистическая оценка неизвестного значения исключительно на основе корреляционно-регрессионного анализа и необходимость задания ряда важных параметров приводит к необходимости убедиться в правдоподобности восстановленных значений.

8) *Resampling метод.* Метод является итеративным и имеет две модификации, которые основаны на построении регрессионных моделей с последующим усреднением полученных оценок для пропущенных значений. Преимуществом данного метода является повторное использование исходных данных, ведь увеличение числа подвыборок позволяет наиболее полно использовать исходную информацию. С другой стороны, объем новой информации уменьшается для каждой новой подвыборки, так как увеличивается вероятность того, что данные элементы выборки были уже выбраны раньше, – это основной недостаток метода вкупе с отсутствием процедур его оптимизации.

9) *Метод кластерного анализа.* Особенность метода – его применение не опирается на какую-либо вероятностную модель, но при этом оценить его свойства в статистических терминах не представляется возможным. Однако данный метод обладает существенным достоинством в виде алгоритмической простоты его реализации, а также он позволяет указать предпочтительный порядок восстановления данных и выявить случаи, когда пропуски не могут быть восстановлены по имеющимся данным.

Обобщая результаты обзора наиболее распространенных методов восстановления пропусков в статистических данных, следует отметить, что оценки для пропущенных значений, как правило, вычисляются по присутствующим данным, что вносит искусственную зависимость между наблюдениями. Кроме того, распределение данных после заполнения будет отличаться от истинного, даже если пренебречь зависимостью, указанной выше. Этот факт особенно нагляден для простых методов заполнения (средневыборочных, по регрессии) и при существенном количестве пропусков.

Вместе с тем с точки зрения первичной необходимости решения задачи кластеризации в условиях наличия пропусков в статистических данных очевидно, что применение методов восстановления данных на предварительном этапе не является экономичным. Кроме того, восстановление пропущенных значений с использованием указанных методов (кроме простейших)

фактически опирается на поиск тех же зависимостей (наряду собственно с кластеризацией) в имеющихся данных. В этой связи в данной работе предлагается способ совмещения процедуры кластеризации статистических данных и обработки пропущенных в них значений. Способ основан на модификации алгоритма конкурентного обучения сети Кохонена для проведения сферической кластеризации [5].

2. Адаптация модифицированного алгоритма конкурентного обучения для работы с неполными статистическими данными

Будем считать, что доступные статистические данные представлены в виде обучающей выборки из N элементов $x(t)$, $t = 1, \ldots, N$, а также что проведена предварительная нормализация значений всех числовых непрерывных входных переменных к отрезку [0, 1]. Каждый элемент данной выборки представляет собой вектор чисел, компоненты которого соответствуют значениям входных переменных решаемой задачи кластеризации и представляют вещественные или целочисленные значения, в зависимости от типа соответствующей входной переменной.

Инициализация первоначальной структуры обучаемой без учителя сети Кохонена (количество нейронов выходного слоя) проводится на основании задаваемого максимально допустимого количества кластеров H. На самом же деле количество сформированных кластеров K к окончанию процедуры кластеризации, как правило, оказывается меньше этого значения. Вектор весов каждого нейрона выходного слоя определяет центр соответствующего ему кластера в пространстве входных переменных размерности n. Начальные значения весов определяются по специальному алгоритму, описание которого дано ниже.

Во время проведения самоорганизации после предъявления очередного вектора x из обучающей выборки нейроны выходного слоя сети соревнуются, и тот нейрон, вектор весов c которого оказывается ближе к предъявленному вектору, объявляется нейроном-победителем, также определяется и его ближайший конкурент, т.е. нейрон, вектор весов которого оказывается ближе к предъявленному вектору, за исключением нейрона-победителя. В качестве меры близости предлагается использовать следующую метрику:

$$d^*(x(t), c_k) = \sum_{i=1}^{n} h_i (x_i - c_{ki})^2 ,$$

где $h_i = 0$, если i-я входная переменная является категориальной, а также соответствующие значения компонент векторов $x(t)$ и c_k не пропущены и не совпадают; $h_i = 1$ – в противном случае. Метрика $d^*(x(t), c_k)$ также масштабируется количеством побед нейрона c_k в прошлом:

$$d(x(t), c_k) = \frac{n_k}{\sum_{k=1}^{H} n_k} d^*(x(t), c_k) ,$$

где n_k – общее количество выигрышей соответствующего нейрона с начала обучения. Победитель получает право уточнить свои веса в направлении вектора x, а ближайший конкурент – в противоположном направлении. Для улучшения сходимости алгоритма коэффициенты сдвига нейрона-победителя и его ближайшего конкурента динамически сокращаются на каждой эпохе обучения по линейному закону.

Таким образом, суть модификации классического алгоритма конкурентного обучения сети Кохонена [6] состоит в определении на каждой итерации как нейрона-победителя, так и его ближайшего конкурента, в коррекции их весов, а также в использовании специальной метрики $d(x(t), c_k)$, позволяющей успешно решать так называемую проблему мертвых нейронов, поскольку расстояние между вектором весов нейрона и предъявляемым вектором модифицируется пропорционально количеству побед данного нейрона в прошлом, и, следовательно, активные нейроны с большим количеством побед штрафуются искусственным завышением этого расстояния.

Суть адаптации данной модификации алгоритма с целью обеспечения его работы при наличии неполных статистических данных состоит в использовании итеративно изменяемой архитектуры самоорганизующейся сети Кохонена, что также учитывается в метрике $d(x(t), c_k)$. При поступлении на вход сети очередного элемента обучающей выборки с пропусками входной слой сети Кохонена фактически редуцируется до состояния наличия в нем только нейронов, соответствующих известным компонентам входного вектора. Также из сети изымаются все связи нейронов входного и выходного слоев, соответствующие пропущенным компонентам очередного элемента обучающей выборки. Далее приводится пошаговое описание предлагаемого алгоритма:

Шаг 1. EpochNumber = 1, $n_k = 1 (k = 1, \ldots, H)$, maxEpochs = 5, $\varepsilon = 0.0001$, $\alpha_w = 0.06$, $\alpha_r = 0.02$, $H = 30$.

Шаг 2. Инициализировать специальным способом (подробнее алгоритм изложен ниже) центры c_k всех кластеров.

Шаг 3. Инициализировать коэффициенты сдвигов α_w нейрона-победителя и α_r нейрона – ближайшего конкурента: $0 \le \alpha_r \le \alpha_w \le 1$.

Шаг 4. Do

Шаг 5. EpochNumber++. Для каждого элемента обучающей выборки $x(t),\ t = 1, \ldots, N$, выполнить шаги 6 – 9.

Шаг 6. Для очередного элемента обучающей выборки $x(t)$ вычислить расстояния до центров всех кластеров с использованием метрики $d(x(t), c_k)$.

Шаг 7. В результате конкуренции определить нейрон-победитель u_w и нейрон-конкурент u_r согласно правилам: $w = \arg\min_k (d(x(t), c_k))$; $r = \arg\min_{k \ne w} (d(x(t), c_k))$.

Шаг 8. Обновить количество выигрышей нейрона-победителя по формуле:

$$n_k = \begin{cases} n_k + 1, \text{если } k = w \\ n_k, \text{если } k \ne w \end{cases}.$$

Шаг 9. Обновить векторы весов нейрона-победителя и нейрона-конкурента по формулам: $c_w = c_w + \alpha_w (x(t) - c_w)$; $c_r = c_r - \alpha_r (x(t) - c_r)$ для известных компонент вектора $x(t)$. Компоненты векторов c_w и c_r, соответствующие пропущенным компонентам вектора $x(t)$, оставить без изменения.

Шаг 10. Уменьшить коэффициенты сдвига для нейрона-победителя и нейрона-конкурента по линейному закону:

$$\alpha_w = \alpha_w - \alpha_w \frac{EpochNumber}{\max Epochs};$$

$$\alpha_r = \alpha_r - \alpha_r \frac{EpochNumber}{\max Epochs}.$$

Шаг 11. While $\frac{1}{H} \sum_{k=1}^{H} \left\| c_k^{(t+1)} - c_k^{(t)} \right\| > \varepsilon$ (т.е. самоорганизацию проводить до тех пор, пока усредненное суммарное значение изменений координат центров кластеров за одну эпоху обучения не станет меньшим заранее определенного значения точности обучения).

Шаг 12. Удалить из сети все нейроны выходного слоя, векторы весов которых c_k оказались за пределами допустимого нормализованного диапазона [0, 1] хотя бы одной входной переменной.

Таким образом, предложенный алгоритм соревновательного обучения сформирует $K \le H$ нейронов, векторы весов $c_k\ (k = 1, \ldots, K)$ которых будут представлять центры искомых сферических кластеров в пространстве входных переменных.

Для проведения начальной инициализации центров кластеров на шаге 2 представленного алгоритма предлагается использовать следующую итеративную процедуру:

Шаг 2.1. Изначально все элементы обучающей выборки помещены в один общий кластер.

Шаг 2.2. Вычислить вариацию значений (разность максимального и минимального значений) отдельно по каждой переменной по элементам, принадлежащим кластеру.

Шаг 2.3. Выбрать переменную с наибольшей вариацией для проведения разбиения кластера (разделения элементов в нем на два дочерних кластера) по этой переменной.

Шаг 2.4. Упорядочить все элементы кластера вдоль выбранной переменной разбиения по возрастанию ее значений и вычислить квадраты расстояний между соседними элементами, используя метрику $d^*(c_j, c_{j+1})$.

Шаг 2.5. Вычислить значение центроида кластера вдоль оси, соответствующей выбранной на шаге 2.3 переменной, по формуле:

$$centroid = \frac{\sum_{i=1}^{N} ds_i}{N}, \text{ где } \quad ds_i = \sum_{j=1}^{i} d^*(c_j, c_{j+1}).$$

Шаг 2.6. Разбить кластер на два новых плоскостью, перпендикулярной выбранной на шаге 2.3 переменной и проходящей через точку, рассчитанное значение ds_i для которой отличается от рассчитанного значения центроида кластера на наименьшую величину.

Шаг 2.7. Вычислить общую ошибку кластеризации на данном шаге для данного кластера как разность суммы расстояний между всеми его элементами и центром кластера и аналогичных сумм, рассчитанных для двух дочерних кластеров. Поместить данный кластер в кучу, используя рассчитанную ошибку кластеризации в качестве ключа.

Шаг 2.8. Удалить из кучи кластер с максимальным значением ошибки кластеризации.

Шаг 2.9. Для обоих его дочерних непустых кластеров выполнить шаги 2.2–2.7.

Шаг 2.10. Повторять шаги 2.8, 2.9 до тех пор, пока не будет сформировано заданное количество кластеров *H*. Рассчитанные центроиды полученных кластеров как точки с координатами, являющимися средними арифметическими

координат точек, принадлежащих кластеру, и составят начальные центры кластеров.

В отличие от инициализации центров кластеров псевдослучайными значениями в классическом алгоритме конкурентного обучения сети Кохонена [6] представленная процедура пытается осмысленно разбивать данные на заданное количество кластеров так, чтобы снизить суммарную ошибку кластеризации для всех кластеров и, вместе с тем, повысить суммарное значение расстояний между их центрами, что в конечном счете способствует уменьшению количества итераций основного рассмотренного алгоритма.

3. Практические результаты

Предложенный адаптированный для работы с неполными статистическими данными алгоритм конкурентного обучения был программно реализован в рамках программной системы FKNOD, созданной автором для платформы .NET Framework v.2.0 в среде Visual C# 2005 и обеспечивающей широкие возможности для решения задач классификации и прогнозирования. Апробация алгоритма проводилась при решении задачи определения индивидуальных доз радиоактивного йода-131 при лечении больных диффузным токсическим зобом (болезнью Грейвса) [7].

Подготовленная специалистами МЛПУ «Городская больница №13» г. Нижнего Новгорода база данных статистики включала 294 пациента с болезнью Грейвса. Средний возраст составил 48 лет (39; 56), длительность заболевания – 5 лет (3; 9), объем щитовидной железы – 27 мл (19.1; 41.0), минимальный объем щитовидной железы – 3.6 мл, максимальный – 185.5 мл. Общее количество входных переменных – 13. Лечебная доза йода-131 назначалась эмпирически и составила 350 МБк (250; 480), минимальная – 72 МБк, максимальная – 1180 МБк. Гипо- и эутиреоз через 3 месяца после введения йода-131 оценивались как положительный результат лечения, гипертиреоз – как отсутствие ожидаемого результата. Особенностью подготовленной базы данных является существенное количество пропусков: после исключения всех записей, содержащих пропущенные значения, размер базы данных сокращался более чем в 3 раза – до 97 записей.

В рамках предложенной для решения данной задачи четырехэтапной процедуры решались задачи прогнозирования для оценки диапазона рекомендуемых доз для конкретного пациента и задача классификации для оценки достоверности положительного или отрицательного исхода лечения в зависимости от назначаемой пациенту дозы. Качество представленного алгоритма кластеризации оценивалось по значению точности решения указанной задачи классификации после отображения полученной структуры кластеров в нечеткую классифицирующую систему Такаги – Сугено – Канга 0-порядка и ее оптимизации с использованием гибридного иммунного алгоритма [8]. Для оценки параметров точности решения задачи классификации применялся метод перекрестной проверки, на каждой итерации которого использовались непересекающиеся и сформированные псевдослучайным способом обучающие и тестовые выборки исходной базы данных статистики с последующим усреднением полученных значений точности решения по всем итерациям.

Сравнение предложенного адаптированного алгоритма проводилось с известными алгоритмами восстановления пропусков в данных – алгоритмом исключения записей с пропусками, замены пропусков средними по столбцу значениями, а также одной из реализаций EM-алгоритма – AutoClass [9]. Все указанные алгоритмы (кроме предложенного адаптированного) запускались до начала основной процедуры кластеризации по классическому алгоритму конкурентного обучения. Замеры времени работы алгоритмов выполнялись на персональном компьютере, оснащенном процессором Intel® Core i7 940 и 6 гигабайтами оперативной памяти, и проводились от начала выполнения процедуры восстановления пропущенных данных до окончания процедуры кластеризации и также усреднялись по итерациям перекрестной проверки.

Полученные результаты (табл.) показывают, что использование наиболее широко распространенных методов восстановления пропусков в данных (исключение пропусков, замена средними по столбцу значениями), особенно в условиях наличия большого количества пропусков, приводит к достаточно посредственным результатам решения задачи классификации, однако естественным преимуществом данных методов является простота реализации и незначительное время работы, оказавшееся немногим меньше, чем у представленного в работе адаптированного алгоритма. Использование EM-алгоритма обеспечило достаточно высокую и сопоставимую с предложенным адаптированным алгоритмом точность решения задачи классификации, однако потребовало значительно большего времени выполнения. Кроме того, несомненным

Таблица

Сравнение эффективности алгоритмов решения задачи кластеризации при неполных статистических данных

Алгоритм	Усредненная точность	Время работы (в секундах)
Исключение пропусков	68 %	5.2
Замена пропусков средними по столбцу значениями	72 %	9.5
EM-алгоритм	85 %	84.1
Адаптированный алгоритм конкурентного обучения	90 %	12.0

преимуществом представленного в работе алгоритма является простота его программной реализации.

Заключение

Представленный в работе адаптированный для работы с неполными статистическими данными алгоритм конкурентного обучения является экономичным по времени и, с одной стороны, не требует выполнения предварительных процедур восстановления неполных статистических данных, не внося при этом новых зависимостей в известные данные, а с другой – является эффективным и легко реализуемым. Практическая эффективность представленного алгоритма показана на примере решения задачи определения индивидуальных доз радиоактивного йода-131 при лечении больных диффузным токсическим зобом, подготовленная база данных статистики для которой отличалась наличием существенного количества пропусков. Реализация алгоритма выполнена в рамках программной системы, предназначенной для использования врачами-специалистами для назначения индивидуальных доз йода-131. Данная программная система уже применяется специалистами отделения радиологии МЛПУ «Городская больница №13» г. Нижнего Новгорода для повышения эффективности лечения.

Список литературы

1. Литтл Р.Дж.А., Рубин Д.Б. Статистический анализ данных с пропусками. М.: Финансы и статистика, 1990.

2. Злоба Е., Яцкив И. Статистические методы восстановления пропущенных данных // Computer Modelling & New Technologies. 2002. Vol. 6. No. 1. P. 51–61.

3. Schafer J., Graham J. Missing data: our view of the state of the art // Psychological Methods. 2002. Vol. 7. No. 2. P. 147–177.

4. Снитюк В.Е. Эволюционный метод восстановления пропусков в данных // Интеллектуальный анализ информации. Межд. конф. Киев, 2006. С. 262–271.

5. Ефимов А.С. Об одном подходе к извлечению нечетких знаний из статистических данных // Технологии Microsoft в теории и практике программирования: Материалы конференции. Н. Новгород: Изд. Нижегородского госуниверситета, 2007. С. 87–90.

6. Хайкин С. Нейронные сети: полный курс. М.: Вильямс, 2006.

7. Iagaru A., McDougall I. Treatment of thyrotoxicosis // Journal of Nuclear Medicine. 2007. Vol. 48. No. 3. P. 379–389.

8. Ефимов А.С. Гибридный иммунный алгоритм оптимизации нечетких систем TSK 0-порядка // Технологии Microsoft в теории и практике программирования: Материалы конференции. Н. Новгород: Изд. Нижегородского госуниверситета, 2009. С. 143–150.

9. Cheeseman P., Kelly J., Self M. et al. AutoClass: a bayesian classification system // Proceedings of the Fifth International Conference on Machine Learning (Ann Arbor, MI). 1988. P. 54–64.

CLUSTERIZATION PROBLEM SOLUTION BY COMPETITIVE LEARNING FOR INCOMPLETE STATISTICAL DATA

A.S. Efimov

A review of modern methods to restore missing values in incomplete statistical data has been presented. A procedure to combine statistical data clustering with the retrieval of missing values based on a modification of the Kohonen network competitive learning algorithm has been proposed. The efficiency of the proposed procedure has been demonstrated by solving a practical problem to determine individual target doses of radioiodine (^{131}I) in the treatment of diffuse toxic goiter.

Keywords: clusterization, competitive learning, missing values in statistical data.