

УДК 519.7:004.8

Шафроненко А. Ю.¹, Волкова В. В.², Бодянский Е. В.³

¹Стажер-исследователь Харьковского национального университета радиоэлектроники

²Старший преподаватель Харьковского национального университета радиоэлектроники

³Д-р техн. наук, профессор Харьковского национального университета радиоэлектроники

АДАПТИВНАЯ КЛАСТЕРИЗАЦИЯ ДАННЫХ С ПРОПУЩЕННЫМИ ЗНАЧЕНИЯМИ

Предложена адаптивная нейросетевая система, позволяющая решать задачу кластеризации данных с пропущенными значениями в on-line режиме с постоянной коррекцией восстанавливаемых элементов таблицы и центроидов кластеров. Введенная нейросистема характеризуется высоким быстродействием и простотой численной реализации.

Ключевые слова: адаптивная нейросетевая система, восстановление пропусков, кластеризация.

ВВЕДЕНИЕ

Задача кластеризации данных является важным элементом общей проблемы Data Mining, а для ее решения

сегодня существует множество подходов и алгоритмов: от сугубо интуитивных и эвристических до строго математических [1–9]. Вместе с тем, во многих задачах Data

Mining, включая, естественно, кластеризацию, исходные таблицы данных «объект-свойство» могут содержать пустые клетки (пропуски), информация в которых по тем или иным причинам отсутствует. Задаче восстановления таких пропущенных наблюдений уделялось достаточно внимание [6, 10–12], при том наиболее эффективными в данной ситуации оказались подходы, основанные на математическом аппарате мягких вычислений (вычислительного интеллекта) [13, 14], и, прежде всего, искусственных нейронных сетей [12, 15–17]. Вместе с тем, известные подходы к восстановлению пропусков и традиционные алгоритмы кластеризации работоспособны лишь в случаях, когда исходная таблица задана априорно и число ее строк или столбцов не изменяется в процессе обработки. В то же время существует достаточно широкий класс задач, когда данные поступают на обработку последовательно в on-line режиме, при этом заранее неизвестно, какой из обрабатываемых векторов-образов может содержать пропуски. При этом процессы восстановления данных и их кластеризации должны протекать одновременно в реальном времени.

ПОСТАНОВКА ЗАДАЧИ

Пусть задана исходная $(N \times n)$ таблица ($N \geq 1, n \geq 1$) «объект-свойство».

Таблица 1, содержащая информацию об N объектах, каждый из которых описывается $(1 \times n)$ – вектором-строкой признаков $\underline{x}_i = (x_{i1}, \dots, x_{ip}, \dots, x_{ij}, \dots, x_{in})$, при этом предполагается, что N_G строк могут иметь по одному пробелу, а $N_F = N - N_G$ заполнены полностью. В процессе обработки таблицы необходимо заполнить пропуски и сформировать m кластеров. При этом требуется, чтобы восстановленные элементы были бы в определенном смысле «наиболее правдоподобны» или «близки» к априори неизвестным закономерностям, содержащимся в таблице, которые могут неизвестным образом изменяться в процессе обработки.

АДАПТИВНЫЙ АЛГОРИТМ ЗАПОЛНЕНИЯ ПРОПУСКОВ

Для начала представим таблицу 1 в виде $(N \times n)$ – матрицы X , в которой отсутствует один элемент x_{kj} или в более общем случае отсутствуют N_G элементов. Предполагается [10], что между столбцами $\vec{x}_j = (x_{1j}, \dots, x_{ij}, \dots, x_{kj}, \dots, x_{Nj})^T$ существует линейная корреляция, с учетом которой и про-

изводится восстановление пропуска с помощью уравнения линейной регрессии

$$\hat{x}_{kj} = w_{j0} + w_{j1}x_{k1} + w_{j2}x_{k2} + \dots + w_{j,j-1}x_{k,j-1} + \dots + w_{jn}x_{kn} = \underline{X}_{kj} w_j, \quad (1)$$

где $w_j = (w_{j0}, w_{j1}, \dots, w_{jn})^T$ – $(n+1)$ – вектор-столбец параметров, подлежащих определению; $\underline{X}_{kj} = (1, x_{k1}, \dots, x_{k,j-1}, x_{k,j+1}, \dots, x_{kn})$ – $(1 \times n)$ вектор-строка признаков k -го объекта без kj -го элемента и с единицей на первой позиции.

Вектор неизвестных параметров w_j может быть найден с помощью стандартного метода наименьших квадратов, для чего из матрицы X следует исключить k -ю строку, j -й столбец, добавить слева столбец, образованный единицами, и на основе полученной $((N-1) \times n)$ – матрицы X_j рассчитать оценки параметров

$$w_j = (X_j^T X_j)^+ X_j^T \vec{x}_j, \quad (2)$$

где $\vec{x}_j = (x_{1j}, \dots, x_{ij}, \dots, x_{k-1,j}, x_{k+1,j}, \dots, x_{Nj})^T$.

Если же пробелы имеются в N_G строках и в разных столбцах, из матрицы X исключаются все эти строки и на основании усеченной $(N_F \times n)$ матрицы n раз вычисляются векторы параметров w_j (2) для всех $j = 1, 2, \dots, n$. Далее с помощью уравнения (1) заполняются все пробелы полученными оценками $\hat{x}_{kj}$.

Если же данные поступают на обработку последовательно строка за строкой, вместо оценки (2) может быть использован рекуррентный метод наименьших квадратов, который, однако, предлагает неизменность во времени всех связей, существующих в матрице данных. Экспоненциально же взвешенный рекуррентный метод наименьших квадратов может быть неустойчив при малых значениях параметра сглаживания.

В связи с этим для on-line анализа данных представляется целесообразным использование адаптивных алгоритмов обучения, имеющих как фильтрующие, так и следящие (для нестационарных ситуаций) свойства [18]. Если в процессе обработки массива данных было проанализировано и восстановлено N строк, то с приходом $(N+1)$ -го наблюдения оценки w_j уточняются с помощью адаптивной рекуррентной процедуры

$$\begin{cases} w_j(N+1) = w_j(N) + r_j^{-1}(N+1)(x_{N+1,j} - \underline{X}_{N+1,j} w_j(N)) \underline{X}_{N+1,j}^T, \\ r_j(N+1) = \alpha r_j(N) + \|\underline{X}_{N+1,j}\|^2, \end{cases} \quad (3)$$

где $0 \leq \alpha \leq 1$ – параметр сглаживания, задающий компромисс между процессами фильтрации наблюдений и слежением за изменяющимися характеристиками данных.

Таблица 1.

	1	...	p	...	j	...	n
1	x_{11}	...	x_{1p}	...	x_{1j}	...	x_{1n}
...	...	...	...	...	...	...	...
i	x_{i1}	...	x_{ip}	...	x_{ij}	...	x_{in}
...	...	...	...	...	...	...	...
k	x_{k1}	...	x_{kp}	...		...	x_{kn}
...	...	...	...	...	...	...	...
N	x_{N1}	...	x_{Np}	...	x_{Nj}	...	x_{Nn}
...	...	...	...	...	...	...	...

Обработку информации в режиме последовательного поступления данных удобно организовать с помощью нейросетевой системы, основными элементами которой являются n -параллельно работающих адаптивных линейных ассоциаторов (ALA) [19], настраиваемых с помощью алгоритма (3). При этом, если в k -й момент времени на вход системы поступает вектор наблюдений $\vec{x}_k = (x_{k1}, x_{k2}, \dots, x_{kin})$ полный или с пропусками, на выходе появляется он же, если $\vec{x}_k$ полный, или его оценка, если в $\hat{x}_k$ содержались пробелы.

АДАПТИВНЫЙ АЛГОРИТМ КЛАСТЕРИЗАЦИИ

Для решения задачи кластеризации в on-line режиме целесообразно воспользоваться самоорганизующейся картой Т. Кохонена [20], имеющей простую архитектуру с прямой передачей информации и кроме нулевого рецепторного слоя содержащей единственный слой нейронов, чаще всего тех же адаптивных линейных ассоциаторов. Каждый нейрон связан с каждым рецептором нулевого слоя прямыми связями и со всеми остальными нейронами поперечными внутрислойными (латеральными) связями. Именно латеральные связи обеспечивают возбуждение одних нейронов и торможение других.

Благодаря такой организации сети, каждый нейрон ALA получает всю информацию об анализируемом векторе-образе и генерирует на своем выходе соответствующий отклик, после чего между нейронами возникает конкуренция, в результате которой определяется единственный нейрон-победитель с максимальным выходным сигналом. Этот сигнал по латеральным связям обеспечивает возбуждение ближайших «соседей» победителя и подавление реакции далеко отстоящих нейронов. Таким образом в процессе конкурентного самообучения формируются группы нейронов, каждый из которых максимальным образом реагирует на образы «своих» кластеров-подобластей входного пространства сигналов.

Рассмотрим простейшую карту Кохонена, имеющую 1D топологию, n рецепторов и m (по числу возможных кластеров) нейронов, каждый из которых характеризуется собственным вектором синаптических весов $w_q^K, q = 1, 2, \dots, m$, определяющих центроид соответствующего кластера.

Каждый нейрон сети в k -й момент времени получает на входы n -мерный входной вектор

$$\tilde{x}_k = \frac{\hat{x}_k^T}{\|\hat{x}_k\|} \quad (4)$$

и генерирует на своем выходе сигнал

$$y_{kl} = w_l^{KT}(k-1)\tilde{x}_k, \quad (5)$$

зависящий от вектора синаптических весов $w_l^K(k-1)$, настроенных по данным, полученным до момента k , на определенную область (кластер) входного пространства. Близкие в смысле используемой метрики входные векторы $\tilde{x}_k$ и $\tilde{x}_i$ могут возбуждать либо один и тот же нейрон w_q^K , либо два нейрона соседа w_q^K и w_{q+1}^K или w_q^K и $w_{q-1}^K, q = 1, 2, \dots, m$.

В основе алгоритма самоорганизации карты Кохонена лежат принципы конкурентного самообучения [19], при этом как и любая другая процедура настройки нейронной сети, работа алгоритма начинается с инициализации синаптических весов сети, которые обычно выбираются с помощью генератора случайных чисел, при этом для каждого из нейронов должно выполняться условие

$$\|w_l^K(0)\| = 1, l = 1, 2, \dots, m. \quad (6)$$

Процедура самоорганизации реализуется в три основных этапа: конкуренции, кооперации и синаптической адаптации и начинается с анализа образа $\tilde{x}_k$, поступающего с рецепторного слоя на все нейроны сети. Для каждого из нейронов вычисляется расстояние

$$D(\tilde{x}_k, w_l^K(k-1)) = \|\tilde{x}_k - w_l^K(k-1)\|, \quad (7)$$

при этом, если входные векторы предварительно нормированы с помощью выражения (4) так, что

$$\|\tilde{x}_k\| = 1, \quad (8)$$

а в качестве расстояния (7) используется евклидова метрика, то мерой расстояния между $\tilde{x}_k$ и $w_l^K(k-1)$ может служить

$$\begin{aligned} D(\tilde{x}_k, w_l^K(k-1)) &= 2(1 - \tilde{x}_k^T w_l^K(k-1)) = \\ &= 2(1 - \cos(\tilde{x}_k, w_l^K(k-1))), \end{aligned} \quad (9)$$

а мерой близости – скалярное произведение

$$-1 \leq \tilde{x}_k^T w_l^K(k-1) = \cos(\tilde{x}_k, w_l^K(k-1)) \leq 1. \quad (10)$$

Далее определяется нейрон-победитель, «ближайший» ко входному образу $\hat{x}_k$ такой, что

$$D(\tilde{x}_k, w_l^K(k-1)) = \min_l D(\tilde{x}_k, w_l^K(k-1)), \quad (11)$$

после чего, временно опуская этап кооперации, можно подстроить синаптические веса сети с помощью модифицированного правила самообучения

$$w_q^K(k) = \begin{cases} \frac{w_q^K(k-1) + \eta(k)(\tilde{x}_k - w_q^K(k-1))}{\|w_q^K(k-1) + \eta(k)(\tilde{x}_k - w_q^K(k-1))\|}, & \text{если } q\text{-й нейрон победил,} \\ w_q^K(k-1) & \text{в противном случае.} \end{cases} \quad (12)$$

Несложно видеть, что процедура (12) реализует принцип «победитель получает все», при этом вектор синаптических весов $w_q^K(k-1)$ нейрона-победителя «подтягивается» ко входному вектору $\tilde{x}_k$ на расстояние, определяемое шагом настройки $\eta(k)$.

Регулирование величины шага обычно производится, исходя из эмпирических соображений, при этом весьма удобной является процедура [21]

$$\eta(k) = r^{-1}(k), r(k) = \alpha r(k-1) + \|\tilde{x}_k\|^2 = \alpha r(k-1) + 1. \quad (13)$$

Несложно заметить структурную схожесть алгоритмов обучения (3) и (12), (13).

Одной из особенностей карты Кохонена является наличие этапа кооперации в процессе самоорганизации, когда нейрон-победитель определяет локальную область топологического соседства, в которой возбуждается не только он сам, но и его ближайшее окружение, при этом более близкие к победителю нейроны возбуждаются сильнее чем удаленные. Эта топологическая область определяется функцией соседства Φ_{ql} , зависящей от расстояния $D(w_q^K(k-1), w_l^K(k-1))$ между победителем $w_q^K(k-1)$ и любым из нейронов сети $w_l^K(k-1)$. Как правило, Φ_{ql} – это ядерная функция симметричная относительно максимума в точке $D(w_q^K(k-1), w_q^K(k-1)) = 0$ и принимающая в ней единичное значение, монотонно убывающая с ростом расстояния и стремящаяся к нулю при $D(w_q^K(k-1), w_l^K(k-1)) = 2$. При этом наиболее часто в качестве функции соседства используются гауссиан, конус, функции Епанечникова, «мексиканская шляпа» и т. п.

Использование функции соседства приводит к модифицированному правилу обучения

$$\begin{cases} w_l^K(k) = \frac{w_l^K(k-1) + r^{-1}(k)\Phi_{ql}(\tilde{x}_k - w_l^K(k-1))}{\|w_l^K(k-1) + r^{-1}(k)\Phi_{ql}(\tilde{x}_k - w_l^K(k-1))\|} \forall l, \\ r(k) = \alpha r(k-1) + 1, \end{cases} \quad (14)$$

реализующему принцип «победитель получает больше». При $\Phi_{ql} = \delta_{ql} = 1$ при $q=l$ и 0 в остальных случаях, вновь приходим к алгоритму (12), обеспечивающему на каждом такте настройку единственного нейрона-победителя $w_q^K(k-1)$.

В принципе, для обучения самоорганизующейся карты можно вообще обойтись без нахождения победителя, а в качестве функции соседства использовать некоторую конструкцию, зависящую от выходных сигналов нейронов (5) так, что

$$\begin{cases} w_l^K(k) = \frac{w_l^K(k-1) + r^{-1}(k)\varphi(y_{kl})(\tilde{x}_k - w_l^K(k-1))}{\|w_l^K(k-1) + r^{-1}(k)\varphi(y_{kl})(\tilde{x}_k - w_l^K(k-1))\|} \forall l, \\ r(k) = \alpha r(k-1) + 1. \end{cases} \quad (15)$$

Поскольку

$$y_{kl} = w_l^{KT}(k-1)\tilde{x}_k = \cos(\tilde{x}_k, w_l^K(k-1)), \quad (16)$$

то наиболее простой такой конструкцией является

$$\varphi(y_{kl}) = \frac{1 + y_{kl}}{2}, \quad (17)$$

удовлетворяющая всем условиям, предъявляемым к ядерным функциям соседства. Тогда окончательно алгоритм обучения карты Кохонена для нахождения центроидов m кластеров имеет вид

$$\begin{cases} w_l^K(k) = \frac{w_l^K(k-1) + r^{-1}(k)\frac{1+y_{kl}}{2}(\tilde{x}_k - w_l^K(k-1))}{\|w_l^K(k-1) + r^{-1}(k)\frac{1+y_{kl}}{2}(\tilde{x}_k - w_l^K(k-1))\|} \forall l, \\ r(k) = \alpha r(k-1) + 1. \end{cases} \quad (18)$$

Нейросетевая архитектура, реализующая процесс кластеризации данных с пропусками, приведена на рис. 1.

Как видно, введенная нейронная сеть образована однотипными нейронами – адаптивными линейными ассоциаторами, настраиваемыми с помощью достаточно простых и очевидных алгоритмов обучения.

ЗАКЛЮЧЕНИЕ

Рассмотрена задача кластеризации данных, содержащихся в таблицах «объект-свойство» и имеющих пропуски, в режиме последовательного поступления этих данных на обработку. Предложена адаптивная нейросетевая система, позволяющая решать эту задачу в on-line режиме с постоянной коррекцией восстанавливаемых элементов таблицы и центроидов кластеров. Введенная нейросистема образована набором адаптивных линейных ассоциаторов, характеризуется высоким быстродействием и простотой численной реализации.

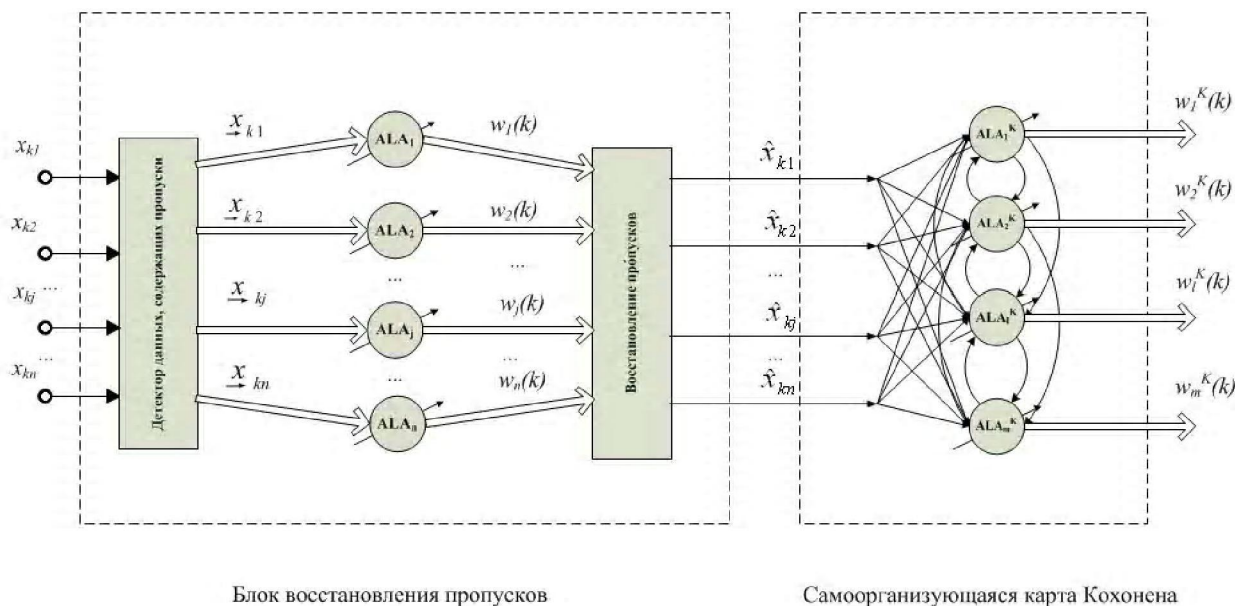


Рис. 1. Искусственная сеть для адаптивной кластеризации данных с пропущенными значениями

СПИСОК ЛІТЕРАТУРИ

1. Jain, A. Data clustering: A review / A. Jain, M. Murty, P. Flynn // ACM Computing Surveys. – 1999. – №3(31). – P. 264–323.
2. Дюк, В. Data Mining / В. Дюк, А. Самойленко. – С. СПб.: Питер, 2001. – С. 368.
3. Friedman, J. The Elements of Statistical Learning. Data Mining, Inference and Prediction / J. Friedman, T. Hastie, R. Tibshirani. – Berlin: Springer, 2003. – P. 552.
4. Xu, R. Survey of clustering algorithms / R. Xu, D. H. Wunsch // IEEE Trans on Neural Networks. – 2005. – №3 (16). – P. 645–678.
5. Han, J. Data Mining: Concepts and Techniques / J. Han, M. Kamber. – Amsterdam: Morgan Kaufman Publ, 2006. – P. 754.
6. Gan, G. Data Clustering: Theory, Algorithms and Applications / G. Gan, Ch. Ma, J. Wu. – Philadelphia: SIAM, 2007. – P. 466.
7. Abonyi, J. Cluster Analysis for Data Mining and System Identification / J. Abonyi, B. Feil. – Basel: Birkhaeuser, 2007. – P. 303.
8. Olson, D. Z. Advanced Data Mining Techniques / D. Z. Olson, D. Dursun. – Berlin: Springer, 2008. – P. 180.
9. Xu, R. II. Clustering / R. Xu, D. C. Wunsch. – Hoboken, N. J.: John Wiley & Sons, Inc., 2009. – P. 341.
10. Загоруйко, Н. Г. Эмпирические предсказания. – Новосибирск: Наука, 1979. – С. 120.
11. Gorban, A. Principal Manifolds for Data Visualization and Dimension Reduction. Lecture Notes in Computational Science and Engineering, Vol. 58 / A. Gorban, B. Kegl, B. Wunsch, A. (Eds.) Zinovyev. – Berlin-Heidelberg – New York: Springer, 2007. – P. 330.
12. Marwala, T. Computational Intelligence for Missing Data Imputation, Estimation, and Management / Marwala T. // Knowledge Optimization Techniques. – Hershey – New York: Information Science Reference, 2009. – P. 303.
13. Maimon, O. Soft computing for Knowledge Discovery and Data Mining / O. Maimon, L. Rokach. – New York: Springer-Verlag, 2007. – P. 448.
14. Rutkowski, L. Computational Intelligence Methods and Techniques / L. Rutkowski. – Berlin-Heidelberg: Springer-Verlag, 2008. – P. 514.
15. Bishop, C. M. Neural Networks for Pattern Recognition. – Oxford: Clarendon Press, 1995. – P. 482.
16. Gorban, A. N. Neural network modeling of data with gaps / A. N. Gorban, A. A. Rossiev, D. C. Wunsch II // Радіоелектроніка, інформатика, управління. – 2000. – №1 (3). – С. 47–55.
17. Tkacz, M. Artificial neural networks in incomplete data sets processing / In: Eds. M. A. Klopotek, S. T. Wierczhon, K. Trojanowski // Intelligent Information Processing and Web Mining. – Berlin-Heidelberg: Springer – Verlag, 2005. – P. 577–583.
18. Bodyanskiy, Ye. An adaptive learning algorithm for a neuro-fuzzy network / Ye. Bodyanskiy, V. Kolodyazhniy, A. Stephan // Ed. B. Reusch «Computational Intelligence. Theory and Applications». – Berlin-Heidelberg – New York: Springer, 2001. – P. 68–75.
19. Haykin, S. Neural Networks. A Comprehensive Foundation / Haykin, S. – Upper Saddle River, N. J.: Prentice Hall, 1999 – P. 842.
20. Kohonen, T. Self-Organizing Maps / T. Kohonen. – Berlin: Springer-Verlag, 1995. – P. 362.
21. Бодянский, Е. В. Искусственные нейронные сети: архитектуры, обучение, применения / Е. В. Бодянский, О. Г. Руденко. – Х.: ТЕЛІТЕХ, 2004. – С. 372.

Стаття надійшла до редакції 06.05.2011.

Шафроненко А. Ю., Волкова В. В., Бодянский С. В.

АДАПТИВНА КЛАСТЕРИЗАЦІЯ ДАНИХ З ПРОПУЩЕНИМИ ЗНАЧЕННЯМИ.

Запропонована адаптивна нейромережева система, що дозволяє вирішувати задачу кластеризації даних з пропущеними значеннями в реальному часі з постійним коригуванням відновлених елементів таблиці і центрів кластерів. Впроваджена нейронна система характеризується високою швидкістю і простою чисельної реалізації.

Ключові слова: адаптивна нейромережева система, відновлення пропусків, кластеризація.

Shafronenko A. Yu. Volkova V. V., Bodyanskiy Yev. V.

ADAPTIVE CLUSTERING WITH MISSING VALUES

In this paper the adaptive neural network system that solves the clustering problem of data with gaps is proposed. This system allows to process the data in the on-line mode with a constant correction of recoverable table's elements and centers of clusters. A proposed neural system has high speed and simple numerical realization.

Key words: adaptive neural network system, the restoration of gaps, clustering.