УДК 004.032.26

Є. В. Бодянський, А. О. Дейнеко, П. Є. Жернова,
О. В. Золотухін, Я. В. Хаустова,
Харківський національний університет радіоелектроніки,
кафедра штучного інтелекту

ПОСЛІДОВНЕ ЯДЕРНЕ НЕЧІТКЕ КЛАСТЕРУВАННЯ ВЕЛИКИХ МАСИВІВ ДАНИХ НА ОСНОВІ ГІБРИДНОЇ СИСТЕМИ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ

Запропоновано архітектуру та методи самонавчання гібридної нейрофаззі системи обчислювального інтелекту для кластерування даних за умов, коли кластери, що формуються, можуть мати довільну форму і взаємно перетинатися. В основу запропонованої системи покладено нечітку узагальнену регресійну нейронну мережу та нейро-фаззі кластерувальну мережу Т. Когонена, налаштування яких основано як на лінивому навчанні, так і на навчанні, що ґрунтується на оптимізації.

Ключові слова: гібридна нейро-фаззі система, нейро-фаззі кластерувальна мережа Т. Когонена, кластерування, нечітка узагальнена регресійна нейронна мережа.

The architecture and self-learning method of hybrid neuro-fuzzy systems for big fuzzy clustering in on-line mode are proposed in this paper. The architecture of proposed system represents the hybrid of the fuzzy general regression neural network and clustering self-organizing network. During a learning procedure in on-line mode, the proposed system tunes both its parameters and its architecture. For tuning of membership functions parameters of neuro-fuzzy system the method based on competitive learning is proposed. The hybrid neuro-fuzzy system tunes its synaptic weights, centers and width parameters of membership functions.

Key words: self-learning method of hybrid neuro-fuzzy systems, big data, fuzzy clustering, clustering self-organizing network, fuzzy general regression neural network.

Вступ

Сьогодні для розв’язання широкого кола задач Data Mining (DM) все частіше використовують методи та засоби обчислювального інтелекту (Computational Intelligence, CI) і, передусім нейро-фаззі системи, завдяки їхнім універсальним апроксимувальним властивостям, прозорості та інтерпретовності отриманих результатів. Особливе місце в DM займають задачі кластерування, складність розв’язання яких пов’язана з високим рівнем апріорної невизначеності та необхідністю використовувати підходи, основані на самонавчанні.

І хоча вже розроблено безліч різноманітних методів кластерування, переважна їх більшість обробляє інформацію за припущення, що весь масив вхідних даних сформовано заздалегідь, його обсяг не змінюється під час аналізу, а кожне спостереження-вектор ознак може оброблятися багаторазово. Зрозуміло, що саме це апріорне припущення унеможливлює застосування традиційного підходу до кластерування в ситуаціях, коли дані надходять на опрацювання послідовно у вигляді потоку інформації або їх обсяг настільки великий (Big Data), що багаторазовий аналіз кожного спостереження просто неможливий.

Огляд літератури

Якщо дані на опрацювання надходять у вигляді потоку, найактуальнішими є послідовні методи кластерування, в яких спостереження надходять до системи, що самонавчається, одне за одним, але після опрацювання кожне окреме спостереження вторинно вже може не підлягати аналізу. У таких випадках дуже привабливим видається використання кластерувальних нейронних мереж Т. Когонена – SOM [1], що є за суттю засобом послідовної реалізації класичного методу

К-середніх. Апріорно припускають, що класи, які формуються під час кластерування, є лінійно роздільними (не перетинаються) та мають опуклу форму.

У разі перетинання класів на перший план виходять методи нечіткого кластерного аналізу, найвідомішим з них є метод нечітких С-середніх (FCM) Дж. Бездека [2], який за суттю є узагальненням К-середніх на випадок перетинання класів. FCM, подібно до методу К-середніх, реалізує пакетний режим обробляння даних, і хоча відомі "гібриди" FCM та SOM [3, 4], вони також працюють з фіксованим масивом даних, багаторазово опрацьовуючи кожне спостереження.

Якщо класи, що створюються, мають довільну форму, з успіхом можна використати ядерні мапи (KSOM) [5-7], що самонавчаються, які побудовані на основі ядер Дж. Мерсера [8] та за суттю є машинами опорних векторів (SVM) [9–11], що самонавчаються. Та хоча SVM є ефективним засобом розв'язання багатьох задач DM, зокрема, природно, і кластерування, вони схильні до так званого "прокльону розмірності", оскільки кількість вузлів у нейронній мережі визначається об'ємом вибірки, що оброблюється.

Тому в задачах ядерного кластерування актуальним з позицій Big Data можна вважати використання підходу, основаного на оцінках Е. Парзена [12], узагальнених регресійних нейронних мережах (GRNN) Д. Шпехта [13], які основані на "лінивому навчанні" за принципом "нейрони в точках даних" [14], та теоремі Т. Кавера [15] про можливість лінійної роздільності класів у просторі підвищеної розмірності. І хоча такий підхід також схильний до "прокльону розмірності" та орієнтований на пакетний режим роботи, він може бути пристосований до опрацювання онлайн великих масивів інформації на основі гібридизації нейро-фаззі підходу й еволюційних систем обчислювального інтелекту [16–19], що дає змогу під час самонавчання "вирощувати" архітектуру, найкраще орієнтовану на розв'язання певної задачі ядерного кластерування даних, які утворюють класи довільної форми.

Отже, враховуючи все зазначене вище, метою цієї роботи є розроблення архітектури та методів самонавчання гібридної нейро-фаззі системи для кластерування потоків даних за умов, коли кластери мають довільну форму та перетинаються. Синтезовані методи самонавчання мають бути достатньо простими з обчислювального погляду та розв'язувати задачі потокового опрацювання великих масивів даних за умов можливого перетинання кластерів довільної форми.

Архітектура ядерної нечіткої кластерувальної системи

Запропонована ядерна нечітка кластерувальна система містить чотири шари опрацювання інформації; перший прихований шар фазифікування входового простору, другий прихований шар агрегування, третій прихований шар обчислення центроїдів кластерів у просторі підвищеної розмірності та четвертий вихідний шар для обчислення рівнів належності.

На вхідний нульовий шар системи подаються $(n\times 1)$-вимірні вектори входових сигналів-образів $x(k)=(x_1(k),...,x_i(k),...,x_n(k))^T$ (де $k=1,2,...,N,...$ – поточний дискретний час), попередньо оброблені, так що $-1\le x_i(k)\le 1$, $\frac{1}{N}\sum_{k=1}^{N}x_i(k)=0$. Потік даних, що надходить, повинен бути розділений на m кластерів довільної форми, що, ймовірно, перетинаються. Перший прихований шар містить nh (по h на кожний вхід) функцій належності $m_{li}(x_i)$, $i=1,2,...,n;\ l=1,2,...,h$ та виконує фазифікування входового простору. Кількість функцій належності на кожному вході $h>n$ задається або апріорно, або визначається під час самонавчання системи. Другий прихований шар здійснює агрегування рівнів належності, розрахованих у першому шарі, і складається з h блоків множення. Отже, перші два приховані шари системи, що розглядається, за архітектурою збігаються з відповідними шарами відомих нейро-фаззі систем Такаґі–Суґено–Канґа, Ванґа–Менделя та ANFIS [9]. Основна відмінність системи, що розглядається, від зазначених полягає в тому, що навчання останніх пов'язано з налаштуванням синаптичних ваг за допомогою тих або інших алгоритмів оптимізації, а налаштування системи, яку ми пропонуємо, основане на "лінивому навчанні" та зводиться до налаштування центрів функцій належності. Доцільно зазначити, що відомі нейро-фаззі

системи близькі за природою до радіально-базисних нейронних мереж, а запропонована система є нечітким аналогом GRNN, тобто узагальненою регресійною нейро-фаззі системою.

Отже, якщо на вхід нейро-фаззі системи подано векторний сигнал $x(k)$, елементи першого прихованого шару обчислюють рівні належності $0 \le m_{li}(x_i(k)) \le 1$. Як функції належності найчастіше використовують одновимірні ґавсіани вигляду

$$m_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li}(k))^2}{2s_i^2} \right),$$

де $c_{li}(k)$ – параметр центра l-ї функції належності на i-му вході, що налаштовується в процесі самонавчання; s_i^2 – параметр ширини.

У другому прихованому шарі обчислюються агреговані значення $\prod_{i=1}^{n} m_{li}(x_i(k)) = j_l(k)$. Якщо на вхід системи подано $(n \times 1)$-вимірний вектор $x(k)$, на виході другого прихованого шару з'являється $(h \times 1)$-вимірний образ підвищеної розмірності $j(k) = (j_1(k), ..., j_l(k), ..., j_h(k))^T$. Надалі нечіткому кластеруванню підлягає послідовність $j(1), j(2), ..., j(N), ...$ Третій прихований та четвертий вихідний шари системи, що розглядається, утворюють двошарову нечітку кластерувальну мережу Т. Когонена, в першому шарі якої відновлюються центроїди кластерів $c_1^K, ..., c_j^K, ..., c_m^K \in R^h$, $j = 1, 2, ..., m$, а у вихідному – рівні належності $u_j(k)$ кожного вектора $j(k)$ до кожного кластера з центроїдом c_j^K.

Самонавчання ядерної нечіткої кластерувальної системи

Процес самонавчання системи, що розглядається, складається з налаштування функцій належності першого прихованого шару на основі "лінивого навчання" і методів еволюційних систем і налаштування центроїдів кластерів, що формуються у третьому прихованому шарі.

Процес налаштування функцій належності першого прихованого шару задля наочності проілюструємо на прикладі двовимірного вхідного вектора $x(k) = (x_1(k), x_2(k))^T$, $k = 1, 2, ...$, кількість функцій належності на кожному вході $h = 5$. Отже, кількість центрів $c_{li}(k)$, що налаштовуються, визначається значенням $nh = 10$.

У найпростішому випадку "лінивого навчання" припускається $c_1(0) = x(1), c_2(0) = x(2), ..., c_l(0) = x(l), ..., c_5(0) = x(5)$, а подальше корегування центрів $c_l(k)$ у міру надходження даних до системи здійснюється згідно за технікою, прийнятою у GRNN.

У разі послідовного опрацювання даних зручнішим видається підхід, коли початкові положення центрів функцій належності $c_{li}(0)$ рівномірно розподілені за координатними осями x_1, x_2, а відстань між ними визначається співвідношенням

$$\Delta(0) = \frac{x_{i\max} - x_{i\min}}{h-1} = \frac{2}{h-1} = 0.5$$

для $-1 \le x_i(k) \le 1$.

Отже, у випадку багатовимірного вектора входів $x(k) \in R^n$, центри $c_{li}(0)$ рівномірно розподіляються по осях гіперкуба $[-1, 1]^n$.

Нехай на вхід системи подане перше спостереження $x(1) = (x_1(1), x_2(1))^T$. Далі на кожній із осей є центри – "переможці" (аналоги нейронів – "переможців" у SOM) $c_{li}^*(0)$ найближчих до $x_i(1)$ в сенсі відстані

$$d_{li} = |x_i(1) - c_{li}(0)|,$$

тобто

$$c_{li}^*(0) = \arg\min\{d_{1i}, d_{2i}, ..., d_{hi}\}.$$

Пошук центра-“переможця” є за суттю реалізацією процесу конкуренції за Т. Когоненом [19] з тією лише відмінністю, що “переможці” по різним осях можуть належати функціям належності з різними індексами l. Далі ці “переможці” підтягуються до компонент вхідного сигналу $x_i(1)$ згідно з WTA – правилом самонавчання, яке можна записати у вигляді

$$c_{li}(1)=\begin{cases}c_{li}^{*}+h_{li}(k)(x_i(1)-c_{li}^{*}(0)) - \textit{для переможця},\\ c_{li}(0) - \textit{для всіх інших}.\end{cases}$$

Для довільних n, h та k WTA-правило самонавчання першого прихованого шару має вигляд

$$c_{li}(k)=\begin{cases}c_{li}^{*}(k-1)+h_{li}(k)(x_i(k)-c_{li}^{*}(k-1)) - \textit{для переможця, якщо } l=1,2,...,h;\ i=1,2,...,n;\\ c_{li}(k-1) - \textit{для всіх інших},\end{cases}$$

де $h_{li}(k)$ – параметр кроку навчання, який монотонно зменшується під час налаштування.

Як вже зазначено вище, здійснюється нечітке кластерування послідовності з $(h\times 1)$-вимірних векторів $j(1), j(2),..., j(N),...$ у третьому та четвертому шарах системи, що розглядається.

У класі процедур нечіткого кластерування найформальнішими з математичного погляду є алгоритми, що основані на цільових функціях та вирішують завдання їх оптимізації за тими чи іншими апріорними припущеннями. Найпоширеніший ймовірнісний підхід, оснований на мінімізації критерію (цільової функції)

$$E(u_j(k), c_j^K)=\sum_{k=1}^{N}\sum_{j=1}^{m}u_j^b(k)\left\|j(k)-c_j^K\right\|^2$$

за обмежень

$$\sum_{j=1}^{m}u_j(k)=1,\ \ 0\le\sum_{k=1}^{N}u_j(k)\le N,$$

де b – невід’ємний параметр фазифікування (фазифікатор), який визначає “розмитість” границь між кластерами.

Ввівши функцію Лаґранжа

$$L(u_j(k), c_j^K, l(k))=\sum_{k=1}^{N}\sum_{j=1}^{m}u_j^b(k)\left\|j(k)-c_j^K\right\|^2+\sum_{k=1}^{N}l(k)(\sum_{j=1}^{m}u_j(k)-1),$$

де $l(k)$ – невід’ємні множники Лаґранжа, і розв’язавши систему рівнянь Каруша–Куна–Таккера, нескладно отримати кінцевий результат у вигляді

$$u_j(k)=\frac{(\left\|j(k)-c_j^K\right\|^2)^{\frac{1}{1-b}}}{\sum_{l=1}^{m}(\left\|j(k)-c_l^K\right\|^2)^{\frac{1}{1-b}}},\ \ c_j^K=\frac{\sum_{k=1}^{N}u_j^b(k)j(k)}{\sum_{k=1}^{N}u_j^b(k)},\ \ l(k)=-((\sum_{l=1}^{m}b\left\|j(k)-c_l^K\right\|^2)^{\frac{1}{1-b}})^{1-b},$$

що збігається за $b=2$ з FCM Дж. Бездека [2], а за $b\to 1$ є близьким до чіткого методу К-середніх.

У випадках, коли дані надходять на опрацювання послідовно у on-line-режимі, можна використати рекурентну версію вигляду

$$u_j(k)=\frac{(\left\|j(k)-c_j^K(k-1)\right\|^2)^{\frac{1}{1-b}}}{\sum_{l=1}^{m}(\left\|j(k)-c_l^K\right\|^2)^{\frac{1}{1-b}}},\ c_j^K(k)=c_j^K(k-1)+h(k)u_j^b(k)(j(k)-c_j^K(k-1)).$$

Легко помітити, що друге співвідношення є за суттю WTM-правилом самонавчання Т. Когонена, у якому співмножник $u_j^b(k)$ відіграє роль функції сусідства.

Висновки

Запропоновано архітектуру та метод самонавчання гібридної нейро-фаззі системи обчислювального інтелекту для кластерування даних за умов, коли кластери, що формуються, можуть мати довільну форму і взаємно перетинатися. В основу системи, яку запропоновано, покладено нечітку узагальнену регресійну нейронну мережу та нейро-фаззі кластерувальну мережу Т. Когонена, налаштування яких основане як на "лінивому навчанні", так і на навчанні, що ґрунтується на оптимізації. Введена процедура самонавчання достатньо проста у числовій реалізації та призначена для обробляння даних, що послідовно в on-line-режимі надходять у систему.

1. Kohonen T. Self-Organizing Maps / T. Kohonen // Berlin: Springer-Verlag. – 1995. – 362 p. 2. Bezdek, J.-C. Pattern Recognition with Fuzzy Objective Function Algorithms [Text] / J. C. Bezdek. – N.Y.: Plenum Press, 1981. – 272 p. 3. Tsao E.C.-K. Fuzzy Kohonen clustering networks [Text] / E.C.-K. Tsao, J. C. Bezdek, J. C. Tsao, N. R. Pal // Pattern Recognition. – 1994. – No. 27. – P. 757–764. 4. Pascual – Marqui R. D. Smoothly distributed fuzzy C-means: a new self-organizing map / R. D. Pascual – Marqui, A.D. Pascual – Montano, K. Kochi, J.M. Caroso // Pattern Recognition. – 2001. – No. 34. – P. 2395–2402. 5. MacDonald D., Fyfe C. Clustering in data space and feature space [Text] : ESANN'2002 Proc. European Symp. on Artificial Neural Networks. Bruges (24-26 April 2002). – Belgium. – 2002. – P. 137–142. 6. Girolami, M. Mercer kernel-based clustering in feature space [Text] / M. Girolami // IEEE Trans. on Neural Networks. – 2002. – Vol. 13. – No. 3. – P. 780–784. 7. Camastra F. A novel kernel method for clustering [Text] / F. Camastra, A. Verri // IEEE Trans. on Pattern Analysis and Machine Intelligence. – 2005. – No. 5. – P. 801–805. 8. Schölkopf, B. Learning with Kernels [Text] / B. Schölkopf, A. Smola // Cambridge M. A.: MIT Press. – 2002. – 648 p. 9. Kacprzyk J. Springer Handbook of Computational Intelligence [Text] / J. Kacprzyk, W. Pedrycz. – Berlin Heidelberg: Springer – Verlag, 2015. – 1634 p. 10. Haykin, S. Neural Networks and Learning Machines [Text] / S. Haykin. – N.Y. :Prentice Hall, 2009. – 1634 p. 11. Cortes C. Support Vector Networks [Texy] / C. Cortes, V. Vapnik // Machine Learning. – 1995. – No. 20. – P. 273–297. 12. Parzen E. On the estimation of a probability density function and the mode / E. Parzen // Ann. Math. Statist. – 1962. – No. 38. – P. 1065–1076. 13. Specht, D.F. A general regression neural network [Text] / D.F. Specht // IEEE Trans. on Neural Networks. – 1991. – Vol. 2. – P. 568–576. 14. Zahirniak D. Pattern recognition using radial basis function network. [Text] / D. Zahirniak, R. Chapman, S. Rogers, B. Suter, M. Kabrisky, V. Piati // Proc 6th Ann. Aerospace Application of Artificial Intelligence Conf. – Dayton, OH. – 1990. – P. 249–260. 15. Cover T. M. Geometrical and statistical properties of systems of linear inequali-ties with applications in pattern recognition [Text] / T.M. Cover // IEEE Trans. on Electronic Computers. – 1965. – No. 14. – P. 326–334. 16. Angelov, P. Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems [Text] / P. Angelov // Heidelberg-New York: Springer-Verlag. – 2002. – 211 p. 17. Kasabov N. Evolving Connectionist Systems [Text] / N. Kasabov – London: Springer-Verlag. – 2003 – 307 p. 18. Angelov P. Evolving computational intelligence systems [Text] / P. Angelov, N. Kasabov // Proc. 1st Int. Workshop on Genetic Fuzzy Systems. – Granada, Spain. – 2005. – P. 76–82. 19. Lughofer E. Evolving Fuzzy Systems – Methodologies and Applications [Text] / E. Lughofer. – Studies in Fuzziness and Soft Computing. – Springer-Berlin. – 2011. – 410 p.