© О.В. НИССЕНБАУМ
onissenbaum@rambler.ru

УДК 519.254

АЛГОРИТМ КЛАСТЕРИЗАЦИИ ПОТОКА ДАННЫХ С ИЗМЕНЯЮЩИМИСЯ ПАРАМЕТРАМИ РАСПРЕДЕЛЕНИЯ

АННОТАЦИЯ. На основании динамического EM-алгоритма построен алгоритм кластеризации для потока данных, взвешенных по времени поступления. Алгоритм предназначен для кластеризации данных с нормальным распределением в $\boldsymbol{R}^n$, параметры которого изменяются во времени, что соответствует ситуации в реальных динамических системах, таких как компьютерные системы, сети связи и т.п. Хранения обработанных данных не требуется, алгоритм эффективно вычислим, может применяться в системах реального времени. Приведены данные вычислительного эксперимента (на имитационной модели потока с нормальной плотностью распределения кластеров), показавшие более высокое качество работы по сравнению с алгоритмом, в котором не используются весовые коэффициенты от времени, с точки зрения доли неверно распознанных точек и точности определения параметров рассчитываемых кластеров.

SUMMARY. The article contains a clustering algorithm for time-weighted data streams based on the dynamic EM-algorithm. This algorithm can be used for clustering data with the normal distribution in Rn, the parameters of the distribution undergoing changes over time, which is the case in real dymaniv systems such as computer systems or communication nets. The author offers the results of the computational experiment (based on the imitation model with the normal density of cluster distribution), which prove better quality of the proposed algorithm as to the percent of the erroneously recognized points and precision in cluster parameters description in contrast with the algorithm which does not use the time-weighed factors.

КЛЮЧЕВЫЕ СЛОВА. Алгоритм кластеризации, поток данных, динамические данные, нормальное распределение, системы реального времени.

KEY WORDS. Clustering algorithm, data streams, dynamic data, normal distribution, real-time system.

Введение. Проблема кластеризации потоков данных рассматривается с 80-х годов [1] (хотя соответствующая модель формализована лишь в 1998 году [2]) и в последние годы становится все более актуальной в связи с развитием систем рельного времени, таких как компьютерные и вычислительные системы и сети, сети связи, системы управления производственными процессами, а также с необходимостью автоматического мониторинга таких систем. В 2003 г. Д. Барбара сформулировал три требования к алгоритмам кластеризации потоков данных [3]: 1) data compression and expression of the compressed data; 2) processing new data point in a fast and incremental way; 3) distinguishing outliers quickly and clearly. Публикуется множество работ, посвященных созданию и примене-

нию алгоритммов кластеризации (в основном используются различные модификации алгоритма k-средних) к потокам данных, например [4-7].

В [8] поставлена задача монитринга компьютерной сети или отдельного ее канала в целях отслеживания подозрительной активности. При этом данные поступают постоянно, количество измерений велико, а характеристики системы могут со временем меняться. Задача состоит в разработке алгоритма кластеризации данных, который, во-первых, позволяет обрабатывать динамические данные в режиме реального времени; во-вторых, не требует хранения обработанных данных; в-третьих, учитывает «новые» данные с большими весами, чем «старые».

1. EM-алгоритм для потока данных.

За основу взят динамический EM-алгоритм [9], состоящий в следующем: пусть в некоторый момент времени набор измерений (назовем их *исходными данными*) разбит классическим EM-алгоритмом на множество кластеров. Каждый кластер C_k представлен функцией плотности нормального распределения в пространстве характеристик:

$$\varphi(x|\mu, \Sigma)=\{(2\pi)^d|\Sigma|\exp[(x-\mu)^T\Sigma^{-1}(x-\mu)]\}^{-1/2}, \quad (1)$$

где μ и Σ — координаты центра и ковариационная матрица кластера, рассчитанные согласно соответствующим формулам для нормального распределения [9], $x\in\mathbf{R}^d$ — координаты новой точки, индекс k — номер кластера, для простоты опущен.

Пусть теперь $\{x_1, x_2, ...\}\in\mathbf{R}^d$ — *поток данных*, то есть точки, поступающие в последовательные моменты времени $t_1, t_2, ...$. В момент поступления очередной точки x, она относится к одному из кластеров методом максимального правдоподобия. Вероятность π_k принадлежности точки x к кластеру C_k, содержащему N_k точек, определяется формулой

$$\pi_k = \frac{\eta_k\varphi_k(x|\mu_k,\Sigma_k)}{\sum_{i=1}^{K}\eta_k\varphi_k(x|\mu_k,\Sigma_k)}, \quad (2)$$

где K — количество кластеров, $\eta_k=N_k/(N_1+N_2+...+N_K)$ — доля точек в C_k.

Затем характеристики кластера, к которому отнесена точка, пересчитываются по формулам:

$$\mu_+ = \frac{N\mu_- + x}{N+1}; \quad \Sigma_+ = \frac{N}{N+1}\left(\Sigma_- + \frac{(\mu_- - x)^2}{N+1}\right), \quad (3)$$

где индексы «–» и «+» соответствуют значениям до и после пересчета, индекс k опущен.

Этот алгоритм удовлетворяет первым двум требованиям, но не удовлетворяет третьему, то есть не позволяет учитывать измерения с различными весами.

В работе [10] описана следующая модификация динамического EM-алгоритма, позволяющая создавать новые кластеры и избавляться от старых, потерявших актуальность. Вводится пороговое значение вероятности p, и если для новой точки x все верояности $\pi_i<p$, создается новый кластер с центром в точке x. Если же коэффициент η_k становится меньше другого порогового значения ε, то соответствующий кластер C_k удаляется. Такой алгоритм больше подходит для анализа потоков данных, так как учитывает возможность появления новых класте-

ров и потерю актуальности старыми, но все же как старые, так и недавние измерения оказывают на характеристики кластеров равное влияние.

2. Весовая функция кластера от времени.

Предлагаемый подход состоит в замене параметра N_k в формулах (2) и (3), являющегося количеством точек в кластере C_k, на весовую функцию $W_k(t)$, зависящую не только от количества точек, попавших в кластер, но и от того, как давно были получены эти точки. Этот подход был предложен автором в [11], где весовая функция определялась эвристически и не включала в себя оценку мощности кластера.

Определим теперь эту весовую функцию, исходя из следующих соображений. Располагая достаточным количеством памяти, чтобы хранить все полученные в ходе наблюдений измерения и времени, чтобы пересчитывать характеристики кластеров «с нуля» (напомним, что и то, и другое труднодостижимо, если система наблюдается в режиме реального времени достаточно долго), мы ввели бы для каждой точки кластера отдельный весовой коэффициент $w(t-t_i)$, где $w(t)$ — асимптотически и монотонно убывающая к нулю на $[0,+\infty)$ функция ($w(0)=1$), а $t-t_i$ — время, прошедшее с момента поступления точки x_i.

Тогда координаты центра кластера в момент времени t, в который в кластер поступила новая точка x_+ (с весом $w(0)=1$), вычислимы по формулам: $\mu_-=\Sigma(w(t-t_i)x_i)/\Sigma w(t-t_i)$ — без учета новой точки и $\mu_+=(\Sigma w(t-t_i)x_i+x_+)/(\Sigma w(t-t_i)+1)=(\Sigma w(t-t_i)\mu_-+x_+)/(\Sigma w(t-t_i)+1)$ — с учетом новой точки (сумма берется по всем точкам, вошедшим в кластер до момента t, индекс номера кластера опущен), что приводит нас к формуле (3) при $N=\Sigma w(t-t_i)$. Аналогично, приходим к формуле (3) для ковариационной матрицы кластера с такой же заменой, и вес кластера в момент времени t вычисляется как $W(t)=\sum w(t-t_i)$, где сумма берется по всем точкам, вошедшим в кластер.

Поскольку $w(t)$, по определению — функция, монотонно убывающая к нулю, то $\forall\varepsilon>0\ \exists\Delta>0$: при $t-t_i>\Delta$, $w(t-t_i)<\varepsilon$. При достаточно малом ε это значит, что с некоторого момента времени точка x_i перестает оказывать сколько-нибудь заметное влияние на характеристики кластера, то есть устаревает. При заданном Δ, получим:

$$W(t) = \sum_{t-t_i\le\Delta} w(t-t_i)\,, \qquad (4)$$

и хранимыми данными становятся только $t_i\ge t-\Delta$ — моменты поступления точек за ограниченный отрезок времени. Полученные значения $W_k(t)$ используются при определении принадлежности поступившей точки кластеру по формуле (2) и при пересчете характеристик этого кластера по формулам (3).

3. Обратная экспоненциальная функция веса точки.

Вопрос выбора весовой функции точки $w(\cdot)$ является важным. Необходимо использовать такой ее вид, при котором пересчет по формуле (4) не являлся бы трудоемким, и при этом выполнялись требования монотонного убывания к нулю и $w(0)=1$ (новая точка имеет вес 1). Возможно множество вариантов выбора функции $w(\cdot)$, остановимся подробнее на

$$w(t)=e^{-at},\ t\ge0. \qquad (5)$$

Пусть в моменты $0 < t_1, t_2, \ldots, t_n < t$ в кластер попадали точки. Тогда, обозначив $\Delta_i=t-t_i$, $i=1,2,\ldots,n$, согласно (4) и (5), получим $W(t)=e^{-a\Delta_1}+e^{-a\Delta_2}+\ldots+e^{-a\Delta_n}$.

В момент $t+\Delta t$, при условии, что на отрезке $[t,t+\Delta t]$ точек в этот кластер не попадало, имеем $W(t+\Delta t)=e^{-a(t+\Delta t-t_1)}+e^{-a(t+\Delta t-t_2)}+\ldots+e^{-a(t+\Delta t-t_n)}=e^{-a\Delta t}\ [e^{-a\Delta_1}+e^{-a\Delta_2}+\ldots+e^{-a\Delta_n}]$, т.е.

$$W(t+\Delta t)=e^{-a\Delta t}W(t), \qquad (6)$$

если же в момент времени $t+\Delta t$ очередная точка была отнесена к кластеру, то

$$W(t+\Delta t)= W(t+\Delta t-0)+1=e^{-a\Delta t}W(t)+1. \qquad (7)$$

Формулы (6) и (7) позволяют не только легко пересчитывать весовой коэффициент кластера в любой момент времени, но и, в отличие от общего случая, описанного формулой (4), избавляют от необходимости хранить моменты поступления точек в кластер даже на ограниченном интервале времени.

4. Алгоритм кластеризации потока данных с учетом весовой функции от времени.

Исходные данные кластеризуются классическим EM-алгоритмом, рассчитываются характеристики кластеров μ_k и Σ_k. Весовые коэффициенты кластеров устанавливаются равными количеству точек в них ($W_k(t=0)=N_k$).

При появлении в потоке очередной точки x выполняется следующий алгоритм.

Вход: x — новая точка, μ_k , Σ_k, W_k — характеристки и веса кластеров ($k=1,2,\ldots, K$), Δt — время, прошедшее с момента получения предыдущей точки.

Пересчитать веса кластеров $W_k=e^{-a\Delta t}W_k$ (в соответствии с (6)).

Рассчитать вероятности попадания точки x в кластеры согласно (2), используя W_k вместо N_k. Отнести x к одному из кластеров методом максимального правдоподобия.

Для этого кластера произвести пересчет следующих параметров (индекс опущен):

3.1. центра μ и ковариационной матрицы Σ согласно формулам (3), используя W вместо N;

3.2. весового коэффициента $W=W+1$ (в соответствии с (7)).

Заметим, что при использовании (5) с $a=0$ *ед.вр.*$^{-1}$ разработанный алгоритм совпадает с [9]. Заметим также, что изложенный алгоритм можно расширить возможностью создавать новые кластеры и удалять устаревшие, как в [4], дополнив соответствующими действиями на шагах 1 и 2.

Разработанный алгоритм был реализован на C++. На рис. 1 и 2 приведены примеры результата численного эксперимента. Смоделированы данные из 2-х кластеров с нормальным распределением в $\mathbf{R}^2$, причем местоположение центров кластеров изменялось со временем. В эксперименте на рис. 1 центры кластеров смещались навстречу друг другу, а в эксперименте на рис. 2 центр одного кластера смещался в область, где ранее находился центр другого. Точки модлировались с интервалом в 1 *ед.вр*. Обработка данных производилась двумя алгоритмами: [9] и предложенным, в котором использовалась весовая функция (5) при $a=0{,}05$ *ед.вр.*$^{-1}$. Эллипсами обозначены доверительные области кластеров с вероятностью 0,9 (пунктирной линией — истинные, сплошной черной — полученные в предложенном алгоритме, сплошной серой — в [9]) в момент окончания эксперимента.

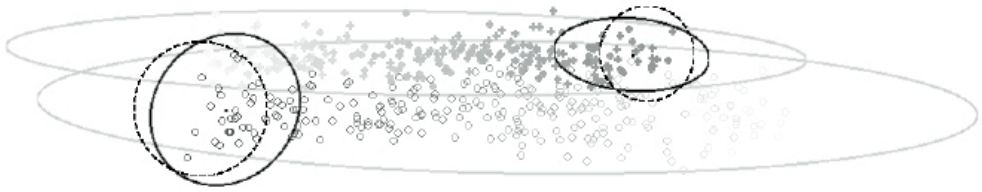

Рис. 1. Данные 1-го кластера обозначены ромбами, 2-го — кружками; интенсивность цвета отражает новизну измерения (темный — новые точки, светлый — старые)

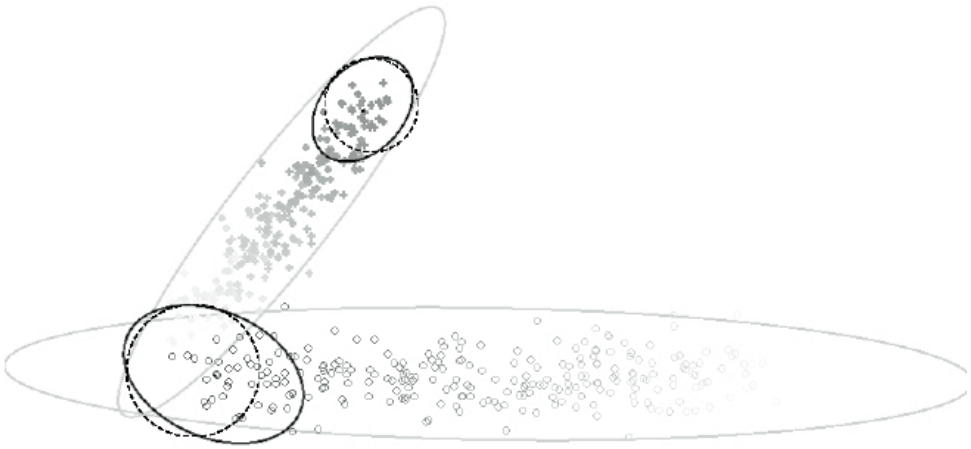

Рис. 2. Данные 1-го кластера обозначены ромбами, 2-го — кружками; интенсивность цвета отражает новизну измерения (темный — новые точки, светлый — старые)

Проведен вычислительный эксперимент, в котором для одних и тех же исходных параметров смоделировано 30 комплектов данных, полученных в двух кластерах с изменяющимся положением центра. Был применен разработанный алгоритм с различными значениями параметра a (в том числе a=0, что соответствует алгоритму из [9]). В момент окончания рассчетов для каждого эксперимента и для каждого значения a были вычислены: z — доля точек, распознанных неверно, l_1 и l_2 — погрешности в определении положения центров первого и второго кластеров соответственно. Их средние и стандартные отклонения по всем тридцати экспериментам приведены в табл. 1.

Таблица 1

a	0	0,01	0,02	0,03	0,04	0,05	0,07
$\bar{z}$	0,0786	0,0396	0,0209	0,0223	0,0319	0,0793	0,1465
$s(z)$	0,0196	0,0135	0,0073	0,0205	0,0550	0,1106	0,1172
$\bar{l}_1$	273,8737	105,9737	50,0275	33,6873	39,7547	83,2458	142,3471
$s(l_1)$	8,3038	10,3710	4,7996	4,3778	48,9153	109,0538	112,2674
$\bar{l}_2$	217,7127	92,5677	42,0568	33,3360	35,1604	75,8002	129,4564
$s(l_2)$	6,9860	12,4656	4,6329	18,3012	47,5823	107,4941	107,9928

Наилучшим диапазоном значений a в данном эксперименте является 0,02-0,04 *ед. вр.*$^{-1}$, качество работы в котором значительно превосходит [9] (см. первый столбец табл. 1). При меньших значениях a принимаются во внимание устаревшие данные, a при больших — работа алгоритма становится нестабильной, так как старые точки имеют слишком маленький вес, и любые ошибки в определении принадлежности точки пагубно отражаются на всей дальнейшей эволюции процесса. Для динамических данных с изменяющимися параметрами распределения данный алгоритм (при определенном выборе a) показывает лучшее качество, чем [9] с точки зрения доли неверно распознанных точек и характеристик кластеров.

Вопрос о выборе параметра a весовой функции (5) остается открытым. Безусловно, он должен зависеть от многих факторов: скорости изменения параметров кластера, частоты поступления точек и т.п. Возможно, для каждого кластера следует определять собственную скорость устаревания, которая, к тому же, изменяется во времени, то есть является динамичеким параметром.

5. Результаты и выводы.

Разработанный алгоритм не требователен к ресурсам (время, память) и пригоден для оперативного мониторинга в больших динамичеких системах, таких как компьютерные системы и сети. Вычислительный эксперимент показал хорошее качество его работы на имитационной модели при надлежащем выборе весовой функции.

СПИСОК ЛИТЕРАТУРЫ

1. Munro, J., Paterson, M. Selection and Sorting with Limited Storage // Theoretical Computer Science. 1980. Pp. 315-323.

2. Henzinger, M., Raghavan, P., Rajagopalan, S. Computing on Data Streams // Digital Equipment Corporation. SRC TN-1998-011, August 1998.

3. Barbara, D. Requirements for clustering data streams // ACM SIGKDD Explorations Newsletter. 2003. Vol. 3. №. 2. Pp. 23-27.

4. Cao, F., Zhou, A. Y. Fast clustering of data streams using graphics processors // Journal of Software. 2007. Vol. 18. №. 2. Pp. 291-302.

5. Zhu, W. H., Yin, J., Xie, Y. H. Arbitrary shape cluster algorithm for clustering data stream // Journal of Software. 2006. Vol. 17. №. 3. Pp. 379-387.

6. Chandrika, J., Ananda Kumar, K.R. Dynamic Clustering Of High Speed Data Streams // International Journal of Computer Science Issues. 2012. Vol. 9. Iss. 2. №. 1. Pp. 224-228.

7. Qian Quan, Chao-Jie Xiao, Rui Zhang. Grid-based Data Stream Clustering for Intrusion Detection // International Journal of Network Security. 2013. Vol. 15. №. 1. Jan. Pp. 1-8.

8. Ниссенбаум О.В., Присяжнюк А.С. Адаптивный алгоритм отслеживания аномальной активности в компьютерной сети на основании характерных изменений оценок альтернирующего потока // Прикладная дискретная математика. 2010. Прил. №3. С. 55-58.

9. Mingzhou Song, Hongbin Wang. Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering // Proceedings of SPIE 5803. 2005. Pp. 174-183.

10. Нестеренко В.А. Эффективный алгоритм кластеризации с нефексированным числом кластеров // М-лы XI Международ. науч.-практич. конф. «Информационная безопасность». Ч.2. Таганрог: Изд-во ТТИ ЮФУ, 2010. С. 102-104.

11. Ниссенбаум О.В., Русаков С.В., Шешняева Е.С. Адаптивный алгоритм кластеризации данных с изменяющимися параметрами распределения // Новые информационные технологии в исследовании сложных структур: м-лы 9-й Российской конференции. Томск: Изд-во НТЛ, 2012. С. 107.

REFERENCES

1. Munro, J., Paterson, M. Selection and Sorting with Limited Storage. *Theoretical Computer Science*. 1980. Pp. 315-323.

2. Henzinger, M., Raghavan, P., Rajagopalan, S. Computing on Data Streams. *Digital Equipment Corporation*. SRC TN-1998-011, August 1998.

3. Barbara, D. Requirements for clustering data streams. *ACM SIGKDD Explorations Newsletter*. 2003. Vol. 3, № 2. Pp. 23-27.

4. Cao, F., Zhou, A. Y. Fast clustering of data streams using graphics processors. *Journal of Software*. 2007. Vol. 18, № 2. Pp. 291-302.

5. Zhu, W.H., Yin, J., Xie, Y.H. Arbitrary shape cluster algorithm for clustering data stream. *Journal of Software*. 2006. Vol. 17, № 3. Pp. 379-387.

6. Chandrika, J., Ananda Kumar, K.R. Dynamic Clustering Of High Speed Data Streams. *International Journal of Computer Science*. 2012. Vol. 9. Issue 2. № 1. Pp. 224-228.

7. Qian Quan, Chao-Jie Xiao, Rui Zhang. Grid-based Data Stream Clustering for Intrusion Detection. *International Journal of Network Security*. 2013. Jan. Vol. 15. № 1. Pp. 1-8.

8. Nissenbaum, O.V., Prisjazhnjuk, A.S. Adaptive algorithm for anomalous network traffic indication based on alternating process. *Prikladnaja diskretnaja matematika. Prilozhenie №3 — Applied Discrete Mathematics. Supplement №3*. 2010. Pp. 55-58. (in Russian).

9. Mingzhou Song, Hongbin Wang. Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering. Proceedings of SPIE 5803. 2005. Pp. 174-183.

10. Nesterenko, V.A. Effective clustering algorithm with the unknown number of clusters [Jeffektivnyj algoritm klasterizacii s nefeksirovannym chislom klasterov]. *M-ly XI Mezhdunarod. nauch.-praktich. konf. «Informacionnaja bezopasnost'». Ch.2* (Proc. of the XI Int. Research Conf. «Information Security». Part. 2). Taganrog, 2010. Pp. 102-104. (in Russian).

11. Nissenbaum, O.V., Rusakov, S.V., Sheshnjaeva, E.S. Adaptive clustering algorithm for the data with changing distribution parameters [Adaptivnyj algoritm klasterizacii dannyh s izmenjajushhimisja parametrami raspredelenija]. *M-ly 9 Rossijskoj konf. «Novye informacionnye tehnologii v issledovanii slozhnyh struktur»* (Proc. of thw 9th Russian Conf. with Int. Participation «New Information Technologies in Complex Structure Research»). Tomsk, 2012. P. 107. (in Russian).