ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2016 Управление, вычислительная техника и информатика № 4 (37)

УДК 519.254
DOI: 10.17223/19988605/37/7

О.В. Ниссенбаум, А.М. Харченко

АДАПТИВНАЯ ОЦЕНКА ВЕСОВОГО КОЭФФИЦИЕНТА В АЛГОРИТМЕ ДИНАМИЧЕСКОЙ КЛАСТЕРИЗАЦИИ ДАННЫХ

Рассматривается адаптивный алгоритм определения коэффициентов затухающей оконной модели динамического EM-алгоритма кластеризации потока данных. 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. Для расчета весовых коэффициентов требуется хранение ограниченного объема данных, алгоритм эффективно вычислим, может применяться в системах реального времени. Приведены данные вычислительного эксперимента (на имитационной модели потока).

Ключевые слова: алгоритм кластеризации; поток данных; системы реального времени; затухающая оконная модель; адаптивный алгоритм.

Интеллектуальный анализ потоков данных на протяжении последних десятилетий вызывает живой интерес исследователей по всему миру. Кластерный анализ – эффективный инструмент анализа данных, закономерными являются многочисленные попытки его применения к потокам данных. В последние годы был предложен ряд алгоритмов динамической кластеризации, учитывающих специфику обработки потоков: высокую размерность, большой объем, постоянно поступающие измерения, которые требуют эффективных, работающих в режиме реального времени алгоритмов [1].

Впервые проблема кластеризации потоков данных, по-видимому, была затронута в 1980-е гг. в [2], а соответствующая модель формализована в 1998 г. [3]. В настоящее время публикуется множество работ, посвященных созданию и применению алгоритмов кластеризации к потокам данных, например [4–16].

Область применения динамической кластеризации данных достаточно широка. Такие алгоритмы востребованы при наблюдении за сложными системами, такими как информационные системы, компьютерные сети, в долговременных популяционных, когнитивных, социологических исследованиях в режиме реального времени [17–21]. Обработка больших объемов информации в режиме реального времени требует эффективных, быстродействующих алгоритмов, а также сжатия данных – необходимо отказаться от хранения всего массива данных (измерений). При этом информация, полученная недавно, зачастую имеет бóльшую значимость, нежели старые данные.

Еще одной важной задачей при разработке алгоритмов машинного обучения является сокращение количества параметров алгоритма (user-defined parameter) путем разработки адаптивных механизмов [1]. Такой механизм для определения параметра затухающего временного окна предложен в данной работе и может быть применен не только для описанного здесь алгоритма кластеризации [22], но и других, таких как DenStream [23].

1. Динамический EM-алгоритм для потока данных, взвешенных по времени

В [22] на основе [24] путем применения затухающей оконной модели построен алгоритм, кластеризующий потоки данных, причем каждой точке соответствует весовой коэффициент $w = w(\Delta t)$, являющийся функцией от времени, прошедшего с момента поступления измерения (точки), а вес кластера $W(t)$ является суммой весов всех входящих в него точек. В качестве весовой функции выбрана $w(\Delta t) = \mathrm{e}^{-a\Delta t}$ ($a \geq 0$), обладающая следующими свойствами:

1) в момент возникновения точка имеет вес, равный 1: $w(0) = 1$;

2) с течением времени вес точки монотонно убывает к 0: $\lim_{\Delta t\to\infty} w(\Delta t) = 0$;

3) если известен вес точки через Δt_1 ед. вр. после ее возникновения, то вес этой же точки еще через Δt_2 ед. вр. рассчитывается по формуле $w(\Delta t_1+\Delta t_2) = w(\Delta t_1)w(\Delta t_2)$;

4) если известен вес кластера в момент времени t и за последующий период времени длительности Δt в кластер не попадало новых точек, то его вес пересчитывается по формуле $W(t+\Delta t) = W(t)w(\Delta t)$.

Такой выбор весовой функции, во-первых, позволяет обрабатывать динамические данные в режиме реального времени; во-вторых, не требует хранения обработанных данных; в-третьих, учитывает «новые» данные с бóльшими весами, чем «старые». Следует отметить, что подобный выбор весовой функции, обеспечивающей затухающее временное окно, является достаточно распространенной практикой и применен независимо, например в [23] (с точностью до основания).

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:

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

где μ и $\sum$ – координаты центра и ковариационная матрица кластера, рассчитанные согласно соответствующим формулам для нормального распределения, $x\in\mathbf{R}^d$ – координаты новой точки, индекс k – номер кластера, для простоты опущен. Весовые коэффициенты кластеров в начальный момент времени устанавливаются равными количеству точек в них.

Пусть теперь $\{x_1, x_2, \ldots\}\in\mathbf{R}^d$ – *поток данных*, т.е. 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 с весом W_k определяется формулой

$$\pi_k = \frac{W_k\varphi_k\left(x\middle|\mu_k, \Sigma_K\right)}{\sum_{i=1}^{K} W_i\varphi_i\left(x\middle|\mu_i, \Sigma_i\right)}. \quad (2)$$

При появлении в потоке очередной точки x выполняется следующий алгоритм.

Вход: x – новая точка, μ_k, $\sum_k$, W_k – характеристики и веса кластеров (k = 1,2,…,K), Δt – время, прошедшее с момента получения предыдущей точки.

1. Пересчитать веса кластеров $W_k = e^{-a\Delta t}W_k$.

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

3. Для этого кластера произвести пересчет центра μ, ковариационной матрицы $\sum$ и веса W:

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

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

Заметим, что весовая функция содержит параметр $a \geq 0$, причем при $a = 0$ разработанный алгоритм совпадает с [24]. Данный параметр влияет на скорость устаревания точки: при его возрастании точка устаревает скорее. Очевидно, он может быть определен отдельно для каждого кластера и зависит от множества факторов: скорости изменения параметров кластера, частоты поступления точек и т.п.

2. Адаптивный алгоритм расчета параметра весовой функции

Назовем точку, полученную в момент времени t_0, незначимой в момент времени $t = t_0+\Delta t$, если ее вес ниже наперед заданного малого значения ε ($0 < \varepsilon << 1$): $w(\Delta t) = e^{-a\Delta t} \leq \varepsilon$. Откуда получаем $a = -\ln \varepsilon/\Delta t$, где Δt – время, за которое точка должна потерять значимость. Определим Δt как время, за которое математическое ожидание точек кластера выйдет за пределы доверительного эллипсоида с доверительной вероятностью p, определенного в момент t_0. Положим, что за время Δt скорость движения центра кластера $\vec{v}$ изменится незначительно, так что можно принять ее постоянной. Тогда, зная направление скорости, рассчитаем r – расстояние от центра кластера до границы доверительного эллипсоида с заданной дове-

рительной вероятностью в этом направлении, и тогда $\Delta t = r/|\vec{v}|$, а оценка для параметра весовой функции будет вычислена как

$$a = -|\vec{v}|\ln \varepsilon / r. \quad (3)$$

Заметим, что при использовании (3) весовая функция W не зависит от единиц измерения времени и расстояния.

Построим оценку для $\vec{v}$, используя $x_1, x_2, \ldots, x_N$ – последние N точек, попавшие в данный кластер, и моменты их поступления $t_1, t_2, \ldots, t_N$. Пусть t_0 – момент времени, когда в кластере была получена точка, предшествующая известным, а μ_0 и Σ_0 – центр кластера и его ковариационная матрица в этот момент. Тогда, поскольку каждая из точек x_i является реализацией случайной величины, которая, в свою очередь, есть сумма нормально распределенной многомерной величины и неслучайного смещения, зависящего от скорости центра кластера и времени, можем заключить, что $x_i = \xi_i + \vec{v}(t_i - t_0)$, где ξ_i – реализация нормально распределенной случайной величины со средним μ_0 и ковариационной матрицей Σ_0, $i = 1,2,\ldots,N$. Тогда, обозначив среднее для N последних точек как μ, имеем

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{\sum_{i=1}^{N} \xi_i}{N} + \vec{v}\frac{\sum_{i=1}^{N}(t_i - t_0)}{N}.$$

Среднее всех ξ_i есть оценка для μ_0, поэтому для $\vec{v}$ можно построить оценку

$$\vec{v} = \frac{N(\mu - \mu_0)}{\sum_{i=1}^{N}(t_i - t_0)}, \quad (4)$$

приняв в качестве μ_0 оценку центра кластера в момент времени t_0.

Для пересчета a предложим следующий подход. В начальный момент времени можно принять $a = 0$, т.е. предположить, что центр кластера не смещается. При попадании в кластер каждой точки накапливать значения μ и знаменателя в (4), а при попадании каждой N-й точки – пересчитывать a согласно (4) и (3). Такой подход позволяет не хранить обработанные точки для алгоритма пересчета a так же, как и для алгоритма кластеризации.

Относительно выбора параметра N можно сделать следующие предположения: N должно быть достаточным для репрезентативности выборки; с другой стороны, при увеличении N снижается частота пересчета коэффициента a. Таким образом, выбор N определяется желаемой частотой пересчета параметра затухания a и ограничен снизу минимально необходимым объемом выборки. Результаты экспериментов, иллюстрирующие вопрос качества оценивания скорости в зависимости от выбора N, приведены в третьем пункте.

В случае возможности выделить для каждого кластера хранилище точек объема N; предложим другой подход к пересчету параметра a: при поступлении в кластер новой точки она добавляется в хранилище вместо самой старой, значение a пересчитывается согласно (4) и (3). В этом случае пересчет параметра затухания происходит при поступлении в кластер каждой точки, и N ограничено лишь выделенным объемом хранилища точек, с одной стороны, и минимальным значением для репрезентативности выборки – с другой.

3. Вычислительный эксперимент

Разработанный алгоритм был реализован на C#. Было проведено несколько серий экспериментов, в каждом из которых моделировалось 9 000 точек в одном кластере, центр которого смещался с постоянной скоростью, а ковариационная матрица оставалась неизменной. Рассчитывались оценки скорости согласно (4) при различных N. В момент поступления каждой точки производилась оценка скорости $\hat{v}$, для скаляра которой рассчитывалась относительная погрешность $\delta = |v - \hat{v}|/v$, были получены выборочные средние $\overline{\hat{v}}$, $\overline{\delta}$ среднеквадратические отклонения σ_v, σ_δ. Перечисленные выборочные характеристики при $|\vec{v}| = 0{,}04$ ед.расст./ед.вр. и постоянной ковариационной матрице $\Sigma = \begin{pmatrix} 400 & 0 \\ 0 & 400 \end{pmatrix}$ приведены в табл. 1.

С увеличением N погрешность, в среднем, уменьшается, как и ее среднеквадратическое отклонение.

Таблица 1

Результаты эксперимента при $\nu = 0{,}04$

N	100	250	500
$\overline{\hat{\nu}}$	0,0724	0,0394	0,0346
σ_ν	0,0196	0,0079	0,0044
$\overline{\delta}$	0,8281	0,1682	0,1544
σ_δ	0,4574	0,1041	0,0820

Траектория скаляра оценки скорости в сравнении с траекторией истинного его значения, для упомянутого эксперимента при $N = 500$ приведен на рис. 1.

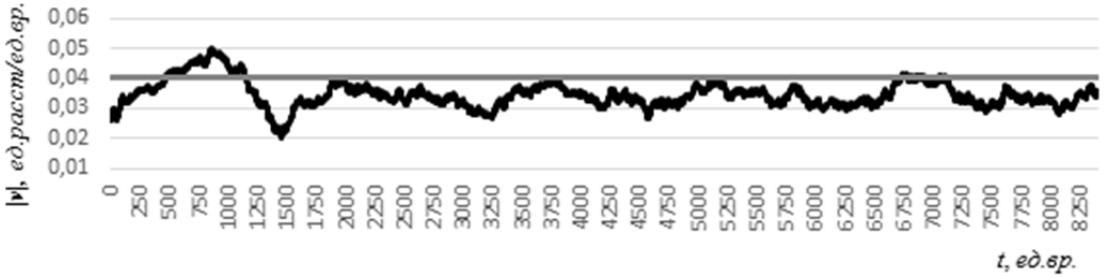

Рис. 1. Траектории скаляров оценки скорости (черная линия) и его истинного значения (серая линия) при $N = 500$

Был проведен вычислительный эксперимент, в котором для одних и тех же исходных параметров было смоделировано 30 комплектов данных, полученных в трех кластерах с нормальным распределением в $\mathbf{R}^2$, причем местоположение центров кластеров изменялось со временем.

Начальное положение центра первого кластера $\mu_1 = (650{,}700)$, ковариационная матрица диагональная, $\sigma_{11}^{\,1} = \sigma_{22}^{\,1} = 700$ (здесь верхний индекс – номер кластера), скорость движения центра $\nu_1 = 7$ ед.расст./ед.вр. Для второго кластера $\mu_2 = (100, 300)$, ковариационная матрица также диагональная, $\sigma_{11}^{\,2} = \sigma_{22}^{\,2} = 400$, $\nu_2 = 5$ ед.расст./ед.вр, для третьего аналогично $\mu_3 = (550, 500)$, $\sigma_{11}^{\,3} = \sigma_{22}^{\,3} = 200$, $\nu_3 = 4$ ед.расст./ед.вр, Направления скоростей задавались случайным образом. Точки моделировались с интервалом в 1/3 ед.вр. с равными вероятностями попадания в каждый из кластеров. Обработка данных производилась предложенным алгоритмом, пересчет параметра a производился согласно (3) и (4). В момент окончания расчетов для каждого эксперимента было вычислено отношение отклонения оцененного центра кластера от истинного к истинному среднеквадратическому отклонению $\Delta\mu_k/\sigma_{11}^{\,k}$. Гистограммы их частот по всем 30 экспериментам приведены на рис. 2.

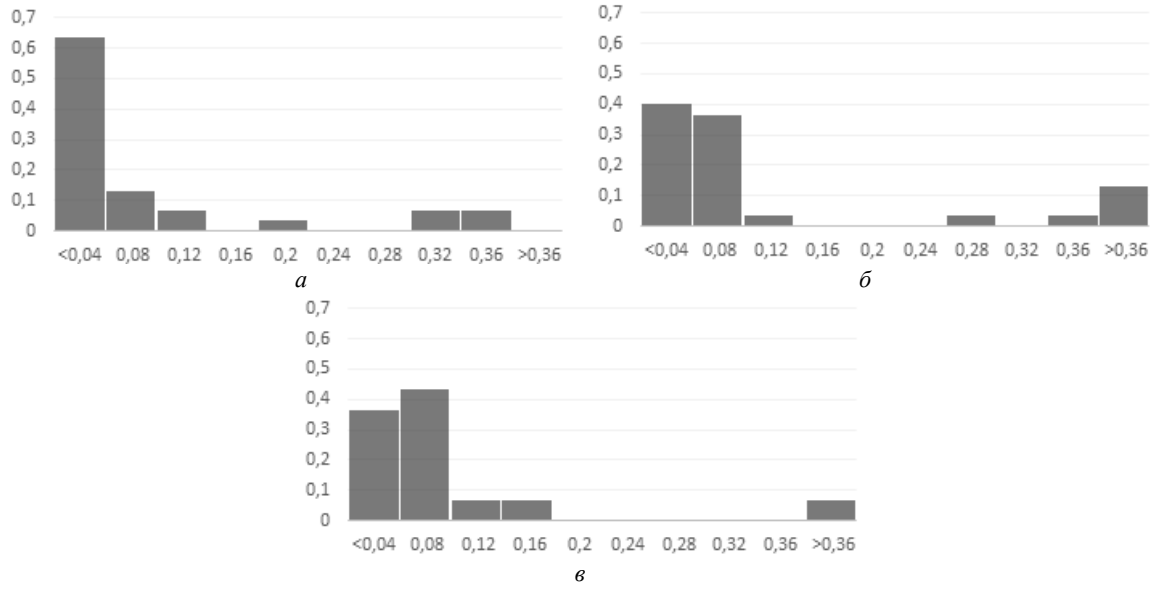

Рис. 2. Гистограммы частот относительных отклонений центров кластеров:
а – первого; *б* – второго; *в* – третьего в момент окончания эксперимента

На рис. 3 приведена визуализация одного из 30 результатов. Эллипсами обозначены доверительные области кластеров с вероятностью 0,9 (пунктирная линия – истинные, сплошная – полученные в предложенном алгоритме) в момент окончания эксперимента.

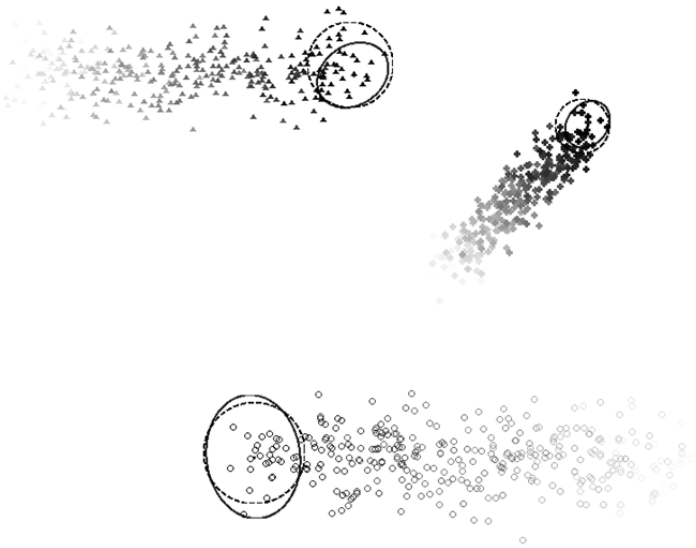

Рис. 3. Точки 1-го кластера обозначены кружками, 2-го – треугольниками,
3-го – крестиками; интенсивность цвета отражает новизну измерения (темный – новые точки, светлый – старые)

Заключение

Разработанный алгоритм нетребователен к ресурсам (время, память) и пригоден для оперативного мониторинга в больших динамических системах, таких как компьютерные системы и сети. Вычислительный эксперимент показал приемлемое качество его работы на имитационной модели.

ЛИТЕРАТУРА

1. Ding S., Wu F., Qian J., Jia H., Jin F. Research on data stream clustering algorithms // Artif Intell Rev. 2015. V. 43. P. 593–600.
2. Munro J., Paterson M. Selection and Sorting with Limited Storage // Theoretical Computer Science. 1980. P. 315–323.
3. Henzinger M., Raghavan P., Rajagopalan S. Computing on Data Streams // Digital Equipment Corporation. SRC TN-1998-011. August 1998.
4. Ackerman M., Martens R., Raupach C., Swierkot K., Lammersen C., Sohler C. StreamKM++: A clustering algorithm for data streams // ACM J. Exper. Algor. 2012. V. 17, No. 1.
5. Aggarwal C.C. A segment-based framework for modeling and mining data streams // Knowl. Inf. Syst. 2010. V. 30, No. 1. P. 1–29.
6. Csernel B., Clerot F., Hebrail G. Data stream clustering over tilted windows through sampling // Proceedings of the Knowledge Discovery from Data Streams Workshop (ECML/PKDD). 2006.
7. Dang X.H., Lee V.C.S., Ng W.K., Ciptadi A., Ong K.-L. An EM-based algorithm for clustering data streams in sliding windows // Proceedings of the 14th International Conference on Database Systems for Advanced Applications. Lecture Notes in Computer Science. Springer, 2009. V. 5463. P. 230–235.
8. Khalilian M., Mustapha N. Data stream clustering: challenges and issues // Proceedings of International Multi Conference of Engineers and Computer Scientists. 2010. P. 566–569.
9. Li Y., Tan B.H. Data stream clustering algorithm based on affinity propagation and density // Advanced Materials Res. 2011. V. 267. P. 444–449.
10. Ong K.L, Li W., Ng W.-K., Lim E.-P. SCLOPE: An algorithm for clustering data streams of categorical attributes // Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery (KDD'04). 2004. P. 209–218.
11. Silva J.A., Hruschka E.R. Extending k-means-based algorithms for evolving data streams with variable number of clusters // Proceedings of the 4th International Conference on Machine Learning and Applications (ICMLA'11). 2011. V. 2. P. 14–19.
12. Cao F., Zhou A.Y. Fast clustering of data streams using graphics processors // Journal of Software. 2007. V. 18, No. 2. P. 291–302.
13. Zhu W.H., Yin J., Xie Y.H. Arbitrary shape cluster algorithm for clustering data stream // Journal of Software. 2006. V. 17, No. 3. P. 379–387.
14. Chandrika J., Ananda Kumar K.R. Dynamic Clustering Of High Speed Data Streams // International Journal of Computer Science Issues. 2012. V. 9, Issue 2, No. 1. P. 224–228.

15. Vorontsov K., Frei O., Apishev M., Romov P., Suvorova M., Yanina A. Non-bayesian additive regularization for multimodal topic modeling of large collections // Proceedings of the 2015 Workshop on Topic Models: Post-Processing and Applications. 2015. P. 29–37.
16. Spirin N., Vorontsov K. Learning to rank with nonlinear monotonic ensemble // International Workshop on Multiple Classifier Systems: Springer Berlin Heidelberg, 2011. P. 16–25.
17. Qian Quan, Chao-Jie Xiao, Rui Zhang. Grid-based Data Stream Clustering for Intrusion Detection // International Journal of Network Security. 2013. V. 15, No. 1. P. 1–8.
18. Daniel A., Sternberg, Kacey Ballard, Joseph L. Hardey at al. The Largest Human Cognitive Performance Dataset Reveals Insights into the Effects of Lifestyle Factors and Aging // Frontiers in Human Neuroscience. 2013. V. 7. Article 292.
19. Zhang X., Sebag M., Germain-Renaud C. Multi-scale real-time grid monitoring with job stream mining // Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'09). 2009. P. 420–427.
20. Shurygin A.M., Strigunova M.S. Prediction of strong earthquakes with the use of scalar product of matrices with a sliding window // Pattern Recognition and Image Analysis (Advances in Mathematiccal Theory and Applications). 2007. V. 17, No. 1. P. 153–162.
21. Ниссенбаум О.В., Присяжнюк А.С. Адаптивный алгоритм отслеживания аномальной активности в компьютерной сети на основании характерных изменений оценок альтернирующего потока // Прикладная дискретная математика. Приложение. 2010. № 3. С. 55–58.
22. Ниссенбаум О.В. Алгоритм кластеризации потока данных с изменяющимися параметрами распределения // Вестник Тюменского государственного университета. 2013. № 7. С. 180–186.
23. Cao F., Ester M., Qian W., Zhou A. Density-based clustering over an evolving data stream with noise // Proceedings of the 6th SIAM International Conference on Data Mining. 2006. P. 328–339.
24. Mingzhou Song, Hongbin Wang. Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering // Proceedings of SPIE 5803. 2005. P. 174–183.

Ниссенбаум Ольга Владимировна, канд. физ.-мат. наук. E-mail: onissenbaum@rambler.ru
Харченко Анастасия Михайловна. E-mail: kharchenkoam.86@gmail.com
Тюменский государственный университет

Поступила в редакцию 4 июня 2016 г.

Nissenbaum Olga V., Kharchenko Anastasia M. (Tyumen State University, Russian Federation).
Adaptive Estimation of Weight Coefficient in a Time-weighted Incremental EM-algorithm for Data Streams.
Keywords: clustering; time-weight; data stream; Gaussian mixture model; damped window model.

DOI: 10.17223/19988605/37/7

Application of data stream clustering is quite general. Applications of data streams include mining data generated by sensor networks, meteorological analysis, stock market analysis, computer network traffic monitoring, long-term population-based, cognitive, real-time sociological studies etc. These applications involve datasets that are far too large to fit in main memory and are typically stored in a secondary storage device. Real-time processing of large volumes of data requires efficient, fast algorithms and data compression. Moreover, recently obtained objects are often more important than the old ones, so several clustering algorithms implement a damped window model with a user-defined decay coefficient. This coefficient is not obvious. An important task in the development of machine learning algorithms is to reduce the number of algorithm parameters (user-defined parameter) through the development of adaptive mechanisms. Such a mechanism for determining the decay coefficient proposed in this paper.

A Gaussian mixture data stream algorithm is constructed with damped window model. Decay coefficient for the object according damped window is $w=w(\Delta t)$ where Δt is time since object arrived from the stream into cluster, weight of the cluster $W(t)$ is sum of all its objects. Recent objects receive higher weight than older ones, and the weights of the objects decrease with time. It is usually to assume $w(\Delta t)=e^{-a\Delta t}$ ($a\geq 0$), with following properties:

1) when arrives, the object weight is equal to one: $w(0)=1$;

2) over time object weight monotonic decreases to zero: $\lim\limits_{\Delta t\to\infty} w(\Delta t) = 0$;

3) if weight of object $w(\Delta t_1)$ after Δt_1 units of time is known, then after more Δt_2 it can be easily recalculates as $w(\Delta t_1+\Delta t_2) = w(\Delta t_1)w(\Delta t_2)$;

4) if weight of cluster $W(t)$ for the instant in time t is known, and during the next time period Δt there was no new objects arrived in this cluster, then weight of this cluster can be recalculates as $W(t+\Delta t) = W(t)w(\Delta t)$.

Let at initial instant in time we have a set of K Gaussian clusters C_k that described by their means μ_k and correlation matrixes $\sum_k$. Weight W_k at initial time defined as a number of objects in cluster C_k. Let $\{x_1, x_2, \ldots\}\in\mathbf{R}^d$ is *data stream*, i.e. set of objects with timestamps $t_1, t_2, \ldots$. When a new object x with timestamp t arrives to the cluster C_k, the following algorithm executes.

Input: new object x, cluster parameters: mean μ_k, correlation matrix $\sum_k$, weight W_k (k=1,2,…,K), time period since last object arrives Δt.

1. Recalculate weights $W_k = e^{-a\Delta t}W_k$ (k=1,2,…,K).

2. Calculate probabilities of belonging x to the k-th cluster $\pi_k = \dfrac{W_k\varphi_k\left(x\middle|\mu_k,\Sigma_K\right)}{\sum_{i=1}^{K} W_i\varphi_i\left(x\middle|\mu_i,\Sigma_i\right)}$. Put x into the most probable cluster.

3. For this cluster (in with x is, index of cluster is dropped) recalculate mean μ, correlation matrix $\sum$ and weight W:

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

where indexes - and + corresponds to the values before and after recalculation.

Note that weight function depends on $a \geq 0$, that may be defined individually for each C_k. We propose the following algorithm to recalculate a separately for each cluster at the time the object arrives into cluster.

Input: $x_1, x_2, \ldots, x_N$ – last N objects putted into cluster, $t_1, t_2, \ldots, t_N$ – their timestamps, μ_0 – cluster mean Σ_0 – cluster correlation matrix.

1. At initial instance of time t_0 adopt $a=0$, i.e. we assume that cluster is not moving.
2. When object x_i with timestamp t_i ($i \leq N$) arrives into the cluster, accumulate variables $\mu = (x_1 + x_2 + \ldots + x_N)/N$ and $\sum_{i=1}^{N}(t_i - t_0)$.
3. If $i=N$ calculate $\vec{v} = \frac{N(\mu - \mu_0)}{\sum_{i=1}^{N}(t_i - t_0)}$ using accumulated variables. Calculate $a = -|\vec{v}|\ln\varepsilon/r$, where r is Euclidean distance between μ_0 and confidence ellipse boundary (confidence probability $p = 1 - \varepsilon$) in direction of $\vec{v}$ vector. Set $t_0=t_N$, assign μ_0 and Σ_0 a corresponding values of cluster at the time t_N and return to step 1.

We perform an experiment using an imitation model of Gaussian mixture clusters with moving centroids. Some results are shown in this article. Proposed algorithm is undemanding to resources (time, memory) and therefore is suitable for real-time monitoring in large dynamic systems, such as computer systems and networks. Quality of decay coefficient adaptation is acceptable as follows from the experimental data.

REFERENCES

1. Ding, S., Wu, F., Qian, J., Jia, H. & Jin, F. (2015) Research on data stream clustering algorithms. *Artif Intell Rev*. 43. pp. 593-600. DOI: 10.1007/s10462-013-9398-7
2. Munro, J. & Paterson, M. (1980) Selection and Sorting with Limited Storage. *Theoretical Computer Science*. pp. 315-323. DOI: 10.1109/SFCS.1978.32
3. Henzinger, M., Raghavan, P. & Rajagopalan, S. (1998) Computing on Data Streams. *Digital Equipment Corporation. SRC* TN-1998-011.
4. Ackerman, M., Martens, R., Raupach, C., Swierkot, K., Lammersen, C. & Sohler, C. (2012) StreamKM++: A clustering algorithm for data streams. *ACM J. Exper. Algor*. 17(1). DOI: 10.1145/2133803.2184450
5. Aggarwal, C.C. (2010) A segment-based framework for modeling and mining data streams. *Knowl. Inf. Syst.* 30(1). pp. 1-29. DOI: 10.1007/s10115-010-0366-0
6. Csernel, B., Clerot, F. & Hebrail G. (2006) Data stream clustering over tilted windows through sampling. *Proceedings of the Knowledge Discovery from Data Streams Workshop (ECML/PKDD).*
7. Dang, X.H., Lee, V.C.S., Ng, W.K., Ciptadi, A. & Ong, K.-L. (2009) An EM-based algorithm for clustering data streams in sliding windows. *Proc. of the 14th International Conference on Database Systems for Advanced Applications*. Lecture Notes in Computer Science. Vol. 5463Springer. pp. 230-235.
8. Khalilian, M. & Mustapha, N. (2010) Data stream clustering: challenges and issues. *Proc. of International Multi Conference of Engineers and Computer Scientists*. pp. 566-569.
9. Li, Y. & Tan, B.H. (2011) Data stream clustering algorithm based on affinity propagation and density. *Advanced Materials Res*. 267. pp. 444-449. DOI: 10.4028/www.scientific.net/AMR.267.444.
10. Ong, K.L, Li, W., Ng, W.-K. & Lim, E.-P. (2004) SCLOPE: An algorithm for clustering data streams of categorical attributes. *Proc. of the 6th International Conference on Data Warehousing and Knowledge Discovery (KDD'04)*. pp. 209-218.
11. Silva, J.A. & Hruschka, E.R. (2011) Extending k-means-based algorithms for evolving data streams with variable number of clusters. *Proc. of the 4th International Conference on Machine Learning and Applications (ICMLA'11)*. 2. pp. 14-19. DOI: 10.1109/ICMLA.2011.67.
12. Cao, F. & Zhou, A.Y. (2007) Fast clustering of data streams using graphics processors. *Journal of Software.* 18(2). pp. 291-302. DOI: 10.1360/jos180291
13. Zhu, W.H., Yin, J. & Xie, Y.H. (2006) Arbitrary shape cluster algorithm for clustering data stream. *Journal of Software.* 17(3). pp. 379-387.
14. Chandrika, J. & Ananda Kumar, K.R. (2012) Dynamic Clustering Of High Speed Data Streams. *International Journal of Computer Science Issues*. 9(2-1). pp.224-228. DOI: 10.1109/ICDE.2009.13.
15. Vorontsov, K., Frei, O., Apishev, M., Romov, P., Suvorova, M. & Yanina, A. (2015) Non-bayesian additive regularization for multimodal topic modeling of large collections. *Proc. of the 2015 Workshop on Topic Models: Post-Processing and Applications.* pp. 29-37. DOI: 10.1145/2809936.2809943
16. Spirin, N. & Vorontsov K. (2011) Learning to rank with nonlinear monotonic ensemble. *International Workshop on Multiple Classifier Systems*. Naples, Italy – June 15 - 17, 2011. Berlin, Heidelberg: Springer. pp.16-25.
17. Qian, Q., Chao-Jie, X. & Rui Zhang. (2013) Grid-based Data Stream Clustering for Intrusion Detection. *International Journal of Network Security*. 15(1). pp. 1-8.
18. Sternberg, D.A., Joseph, Ballard, K. & Hardey, J.L. et al. (2013) The Largest Human Cognitive Performance Dataset Reveals Insights into the Effects of Lifestyle Factors and Aging. *Frontiers in Human Neuroscience*. 7. DOI: 10.3389/fnhum.
19. Zhang, X., Sebag, M. & Germain-Renaud, C. (2009) Multi-scale real-time grid monitoring with job stream mining. *Proc. of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'09).* pp. 420-427. DOI: 10.1109/CCGRID.2009.20

20. Shurygin, A.M. & Strigunova, M.S. (2007) Prediction of strong earthquakes with the use of scalar product of matrices with a sliding window. *Pattern Recognition and Image Analysis (Advances in Mathematiccal Theory and Applications)*. 17(1). pp. 153-162. DOi: 10.1134/S1054661807010191
21. Nissenbaum, O.V. & Prisyagnyuk, A.C. (2010) Adaptive algorithm for anomalous network traffic indication based on alternating process. *Prikladnaya diskretnaya matematika – Applied Discrete Mathematics. Applications.* 3. pp. 55-58. (In Russian).
22. Nissenbaum, O.V. (2013) Clustering algorithm for data streams with changing distribution parameters. *Vestnik Tyumenskogo gosudarstvennogo universiteta*. 7. pp. 180-186. (In Russian).
23. Cao, F., Ester, M., Qian, W. & Zhou, A. (2006) Density-based clustering over an evolving data stream with noise. *Proc. of the 6th SIAM International Conference on Data Mining*. pp. 328-339.
24. Song, M. & Wang, H. (2005) Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering. *Proceedings of SPIE* 5803. pp.174-183. DOI: 10.1117/12.601724