© O. V. NISSENBAUM

onissenbaum@rambler.ru

UDC 519.254

CLUSTERING ALGORITHM FOR DATA STREAMS WITH CHANGING DISTRIBUTION PARAMETERS

ABSTRACT. 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 dynamic 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.

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

Introduction

The problem of data streams clustering has been considered from the 80s [1] (although the corresponding model was formalized only in 1998 [2]) and recently it has become more relevant due to the development of real time systems such as computer and computation systems and networks, communication networks, process control systems, as well as the necessity for automatic monitoring of such systems. In 2003 D. Barbara formulated three requirements for clustering algorithms of data streams [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. A number of works are published, devoted to the development and application of clustering algorithms (mainly various modifications of the k-means algorithm are used) to data streams, such as [4–7].

In [8] the problem of computer network monitoring or its separate channel is set in order to track any suspicious activity. At the same time, the data are received continuously, the number of dimensions is big and system characteristics may vary with time. The task is to develop data clustering algorithm, which, first of all, allows processing dynamic data in real time, and secondly, it requires no storage of the processed data, and thirdly, accounts for the 'new' data with larger value than the 'old' ones.

EM-algorithm for the data stream

Dynamic EM-algorithm is taken as a basis [9], consisting of the following: assume that at some point in time a set of measurements (let's call them original data) is divided by classical EM-algorithm into a set of clusters. Each cluster C_k is represented by the function of density of normal distribution in the feature space:

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

where μ and $\sum$ are coordinates of the center and the covariance matrix of the cluster, calculated according to the corresponding formulas for the normal distribution [9], $x \in \mathbf{R}^d$ are new point coordinates, index k is a cluster number, which is omitted for simplicity.

Now let $\{x_1, x_2, \ldots\} \in \mathbf{R}^d$—*data flow*, that is, the points arriving at successive instants of time $t_1, t_2, \ldots$. Upon arrival of the next point x, it belongs to one of the clusters by maximum likelihood method. Probability π_k of attributing point x to the cluster C_k, containing N_k points, is determined by the formula

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

where K—the number of clusters, $\eta_k = N_k/(N_1 + N_2 + \ldots + N_K)$—fraction of points in C_k.

Then, the characteristics of the cluster to which the classified point is referred, are calculated by the formulas:

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

where the indices + and - correspond to the values before and after the calculation, the index k is omitted.

The algorithm satisfies the first two requirements, but does not satisfy the third, that is, does not allow for measurements with different values.

The given work [10] describes the following modification of dynamic EM-algorithm, which allows creating new clusters and getting rid of the old ones, having lost their relevance. We introduce a threshold limit value probability p, and if for the new point x all probabilities are $\pi_i < p$, a new cluster is built with the center at x. If the ratio η_k becomes smaller than the other threshold limit value ε, then the corresponding cluster C_k is removed. This algorithm is more suitable for data flow analysis, as it considers the possibility of new clusters and loss of relevance of the old ones, but still, both old and latest measurements influence the characteristics of clusters in the same way.

Weighting function of cluster from time

The proposed approach consists in replacing the parameter N_k in formulas (2) and (3) which is the number of points in cluster C_k, on the weight function $W_k(t)$, which depends not only on the number of points in the cluster, but also on how long ago these points were obtained. This approach was suggested by the author in [11], where the weight function was determined heuristically and did not include an assessment of cluster capacity.

We now define this weighting function basing our view on the following considerations. With ample memory to store all the observations received during the measurement and time to recalculate the characteristics of the clusters 'from scratch' (which is rather difficult to achieve if the system is observed in real-time long enough), we have introduced for each point in cluster a separate weighting factor $w(t-t_i)$, where $w(t)$—asymptotically and monotonically decreasing to zero on $[0,+\infty)$ function ($w(0) = 1$), and $t-t_i$—the time elapsed from arrival moment of point x_i.

Then the cluster center coordinates in the time t, in which the cluster receives a new point x_+ (with value $w(0) = 1$) are computable by the formulas: $\mu_- = \Sigma(w(t-t_i)x_i)/\Sigma w(t-t_i)$—excluding the new point and $\mu_+ = (\Sigma w(t-t_i)x_i+x_+)/(\Sigma w(t-t_i)+1) = (\Sigma w(t-t_i)\mu_-+x_+)/(\Sigma w(t-t_i)+1)$—with the new point (the sum is calculated over all points, which were included into the cluster until t, the index of cluster number is omitted), which leads us to the formula (3) with $N = \Sigma w(t-t_i)$. Similarly, we obtain the formula (3) for the covariance matrix of the cluster with the same replacement, and the weight of the cluster at time t is calculated as $W(t) = \sum w(t-t_i)$ where the sum is taken over all points, included in the cluster.

Since $w(t)$, by definition—a function which decreases monotonically to zero, then $\forall\varepsilon > 0\ \exists\Delta > 0$: the $t-t_i > \Delta$, $w(t-t_i) < \varepsilon$. At sufficiently small ε it means that at some point in time point x_i ceases to have any noticeable effect on the characteristics of the cluster, which is obsolete. For the givenΔ, we obtain

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

and stored data are only $t_i \ge t-\Delta$—moments of points arrival for a limited period of time. The obtained values $W_k(t)$ are used for the classification of received point cluster according to formula (2) and under conversion of characteristics of the cluster by the formulas (3).

Inverse exponential function of the point weight

The choice of the weight function $w(\cdot)$ is important. It is necessary to use a form in which the conversion of the formula (4) would not be time-consuming, and thus, meet the requirements of the monotonic decrease to zero and $w(0) = 1$ (new point has a weight of 1). Many variations on the function $w(\cdot)$, we will dwell on

$$w(t) = e^{-at},\ t \ge 0. \tag{5}$$

Suppose that at time $0 < t_1, t_2, \ldots, t_n < t$ the points fell into cluster fall. Then, having denoted $\Delta_I = t-t_i$, $I = 1,2,\ldots,n$, according to (4) and (5), we obtain $W(t) = e^{-a\Delta 1} + e^{-a\Delta 2} + \ldots + e^{-a\Delta n}$. At time $t + \Delta t$, assuming that the interval $[t,t + \Delta t]$ points in the cluster do not fall, we have $W(t+\Delta t) = e^{-a(t+\Delta t-t1)} + e^{-a(t+\Delta t-t2)} + \ldots + e^{-a(t+\Delta t-tn)} = e^{-a\Delta t)}[e^{-a\Delta 1} + e^{-a\Delta 2} + \ldots + e^{-a\Delta n}]$, i.e.

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

if at time $t + \Delta t$ the next point was assigned to the cluster, then

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

Equations (6) and (7) allow easy ways to recalculate the weighting factor cluster at any point in time, and, in contrast to the general case described by formula (4), eliminate the need to store the time points entry in the cluster even in a limited time interval.

The clustering algorithm data flow based on the weighting function of time

Baseline data are clustered according to the classical EM algorithm, the characteristics of the clusters μ_k and $\sum_k$ are calculated. The clusters weighting coefficients are set equal to the number of points in them ($W_k(t = 0) = N_k$).

The entry of the next point x in the stream requires the following algorithm.

Input: x—the new point, μ_k, $\sum_k$, W_k—characteristics and weight of clusters ($k = 1,2,\ldots, K$), Δt—the time, having elapsed since the entry of the previous point.

1. Calculate the weight of clusters $W_k = e^{-a\Delta t}W_k$ (in accordance with (6)).

2. Calculate the probability that a point x is put into clusters according to (2), using W_k instead of N_k. X is attributed to one of the clusters by maximum likelihood method.

3. This cluster requires recalculation of the following parameters (index is omitted):

3.1. center μ and covariance matrix $\sum$ according to (3), using W instead of N;

3.2. a weighting factor W=W+1 (according to (7)).

Note that when using (5) with $a = 0$ *time unit* $^{-1}$, the developed algorithm coincides with [9]. Note also that the above algorithm can be extended by the ability to create new clusters and remove the obsolete ones, as in [4], having added the appropriate actions in steps 1 and 2.

The developed algorithm was implemented in C++. Figures 1 and 2 are the examples of numerical experiments results. The data from 2 clusters were modeled with a normal distribution in $\mathbf{R}^2$, and the location of the cluster centers has changed over time. In the experiment in Fig. 1 the cluster centers were shifted towards each other, and in the experiment in Fig. 2 one cluster center shifted to the area where previously there was the center of another. The points were modeled at intervals of 1 *time unit.* Data processing was performed in two algorithms: [9] and the proposed one, which used a weight function (5) with $a = 0.05$ *time unit* $^{-1}$. Confidence regions are indicated by ellipses clusters with probability 0.9 (dashed line—true, solid black line—obtained in the proposed algorithm, solid gray line— [9]) at the end of the experiment.

Fig. 1. 1-class data cluster are shown as diamonds, 2-class—as circles, color intensity reflects the recency of measurement (dark color—new points, light color—old points)

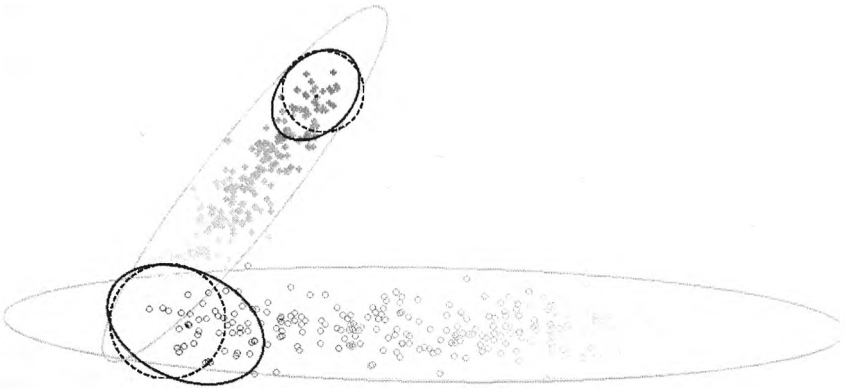

Fig. 2. Description is similar to Fig. 1.

Computational experiment was conducted, in which, for the same input parameters, 30 sets of data were simulated and obtained in two clusters with varying position of the center. The developed algorithm was applied with different values of the parameter a (including $a = 0$, which corresponds to the algorithm in [9]). Upon the end of calculations for each experiment and for each value of a the following parameters were calculated: z—fraction of points incorrectly identified, l_1 and l_2—uncertainty in the position of the first and second clusters centers, respectively. Their average and standard deviations for all thirty experiments are shown in the table below.

Table 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(l1)	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(l2)	6.9860	12.4656	4.6329	18.3012	47.5823	107.4941	107.9928

The best range of values a in the given experiment is 0.02–0.04 *time unit* $^{-1}$, the quality of which far exceeds [9] (see the first column of Table 1). At smaller values of a obsolete data are taken into account, and at large values the algorithm becomes unstable, as the old points have too little weight, and any errors in the classification of points have a negative effect upon the entire future evolution of the process. For dynamic data distribution with varying parameters, the algorithm (for a particular choice *a)* shows better quality than [9] in terms of the proportion of incorrectly identified points and cluster characteristics.

The question about the choice of a weighting function parameter (5) remains open. Certainly, it must depend upon many factors, such as the rate of change of cluster parameters, frequency points, etc. Perhaps for each cluster its own rate of obsolescence should be determined, which also varies in time, being a dynamic parameter.

Results and conclusions

The developed algorithm is undemanding to resources (time, memory) and is suitable for operational monitoring in large dynamic systems, such as computer systems and networks. Computer experiment showed good quality of its work on the simulation model with an appropriate choice of the weight function.

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).