2008 Eighth IEEE International Conference on Data Mining

Comparison of Cluster Representations from Partial Second- to Full Fourth-Order Cross Moments for Data Stream Clustering

Mingzhou (Joe) Song and Lin Zhang
Department of Computer Science
New Mexico State University
Las Cruces, NM 88003, U.S.A.
joemsong@cs.nmsu.edu *and* lin@nmsu.edu

Abstract

Under seven external clustering evaluation measures, a comparison is made for cluster representations from the partial second order to the fourth order in data stream clustering. Two external clustering evaluation measures, purity and cross entropy, adopted for data stream clustering performance evaluation in the past, penalize the performance of an algorithm when each hypothesized cluster contains points in different target classes or true clusters, while ignoring the issue of points in a target class falling into different hypothesized clusters. The seven measures will address both sides of the clustering performance. The represented geometry by the partial second-order statistics of a cluster is non-oblique ellipsoidal and cannot describe the orientation, asymmetry, or peakedness of a cluster. The higher-order cluster representation presented in this paper introduces the third and fourth cross moments, enabling the cluster geometry to be beyond an ellipsoid. The higher-order statistics allow two clusters with different representations to merge into a multivariate normal cluster, using normality tests based on multivariate skewness and kurtosis. The clustering performance under the seven external clustering evaluation measures with a synthetic and two real data streams demonstrates the effectiveness of the higher-order cluster representations.

Keywords: *Data stream clustering, Cluster representation, Cross moment, Gaussian mixture model*

1 Introduction

The richness of cluster representations of unavailable historical data constitutes the repertoire with which incremental merging strategies can play in data stream clustering. Streaming data, partially due to the availability of inexpensive sensor networks or data acquisition instruments, are becoming an increasingly pervasive media of communications with wide applicability. Data stream clustering attempts to detect patterns or clusters from such data. The real-time nature of data streams makes it essential to represent historical data accurately and compactly. Traditional data mining techniques are mainly built on the concept of persistent data sets that are finite, static, and convenient to process in memory. Conversely, a data stream, consisting of temporally ordered points, is transient, continuous, and time-stamped [2]. Thus, data stream clustering aims at separating incoming data, by computation in limited memory, into distinct groups without using historical data, requiring a compact and efficient intermediate form to store summary statistics of past data. A typical data stream clustering algorithm has the following characteristics: a collection, $\mathscr{C}_k$, of summary statistics is constructed to represent cluster k from historical data; if an incoming point x or a sequence of points is determined to be in cluster k, summary statistics $\mathscr{C}_k$ will be updated with x or the sequence; if x or the sequence is determined to be not in any existing cluster, a new cluster that contains only x or the sequence will be created. To simplify the terminology, we define two terms, cluster and class, only for use in this paper: cluster refers to the partitions found by an algorithm, and class to the partitions given in the ground truth.

Existing data stream clustering methods have used up to 2nd-order cluster representations. The BIRCH algorithm [13] creates an in-memory hierarchical data structure called clustering feature (**CF**) tree, in which each node is a cluster and its $\mathscr{C}_k$ includes a mean, the variances of each dimension, and a cluster size. The STREAM algorithm clusters each chunk of the data using the LocalSearch algorithm [11], where $\mathscr{C}_k$ contains a weighted center. CluStream is a flexible framework with online and offline components [1]. The online component extends the CF node in BIRCH by including an additional weight in $\mathscr{C}_k$. The offline component performs global clustering based on all historical sum-

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.143

mary statistics $\{\mathscr{C}_k\}$. DenStream [4], a density-based clustering algorithm, maintains and distinguishes core micro-clusters from outlier micro-clusters for outstanding points. Each core micro-cluster as $\mathscr{C}_k$ consists of a weight, a center and a radius. In the Single Pass K-Means algorithm [3], $\mathscr{C}_k$ consists of a mean, a covariance matrix and a sample size. So far as we know, all previous approaches represent a historical cluster by up to the 2nd-order statistics, corresponding to an ellipsoidal geometry. This paper both examines up to 4th cross moments for cluster representations, enabling the summarization of a cluster by asymmetry and peakedness, and forms strategies to merge two dissimilarly shaped clusters into a multivariate normal one.

Two external clustering evaluation (**ECE**) measures – purity and (cluster-based) cross entropy – have been used for data stream clustering performance evaluation in the past. Both reward the performance when a cluster contains many points of the same class, while ignoring the issue of points in one class falling into different clusters. With five additional ECE measures – class-based cross entropy, homogeneity, completeness, V-Measure, and VI – that address both sides, a comparison of cluster representations from the partial 2nd order to the 4th order indicates the consistent advantage of the 4th-order cluster representation. We evaluated the clustering performance as a function of window size under the seven ECE measures with both synthetic and real data sets to demonstrate the effectiveness of the proposed higher-order cluster representation.

2 Seven External Clustering Evaluation Measures as a Function of Memory Size

External clustering evaluation assesses the performance of a clustering algorithm by comparing the cluster labels with the class labels, providing more reliable [5] performance evaluation than the internal clustering evaluation, when ground truth is available. We will inspect the performance of a data stream clustering algorithm by observing an ECE measure as a function of memory limit or streaming data window size, in order to know whether its performance converges to the theoretical limit.

Let C be a random discrete variable taking value k in $\{1,\ldots,K\}$, denoting the cluster label of a point. Similarly, we define C^*, k^*, and K^* for the class label of a point. Let n be the sample size and n_{k,k^*} the number of points in both cluster k and class k^*. We call $n_{k,\bullet}$ the number of points in, or the size of, cluster k, and analogically $n_{\bullet,k^*}$ for class k^*. Six ECE measures to be defined use information-theoretical concepts, which we summarize here. The entropies of C and C^* are given by

$$H(C) = -\sum_{k=1}^{K} \frac{n_{k,\bullet}}{n} \log \frac{n_{k,\bullet}}{n} \tag{1}$$

and

$$H(C^*) = -\sum_{k^*=1}^{K^*} \frac{n_{\bullet,k^*}}{n} \log \frac{n_{\bullet,k^*}}{n}. \tag{2}$$

The entropy is an uncertainty index of a random variable. The conditional entropies of C given C^* and vice versa are

$$H(C|C^*) = -\sum_{k^*=1}^{K^*} \frac{n_{\bullet,k^*}}{n} \sum_{k=1}^{K} \frac{n_{k,k^*}}{n_{\bullet,k^*}} \log \frac{n_{k,k^*}}{n_{\bullet,k^*}} \tag{3}$$

and

$$H(C^*|C) = -\sum_{k=1}^{K} \frac{n_{k,\bullet}}{n} \sum_{k^*=1}^{K^*} \frac{n_{k,k^*}}{n_{k,\bullet}} \log \frac{n_{k,k^*}}{n_{k,\bullet}}. \tag{4}$$

The conditional entropy is an uncertainty index of one random variable given another. The mutual information between C and C^* is

$$I(C,C^*) = I(C^*,C) = \sum_{k=1}^{K} \sum_{k^*=1}^{K^*} \frac{n_{k,k^*}}{n} \log \frac{n_{k,k^*}}{n_{k,\bullet} n_{\bullet,k^*}}, \tag{5}$$

which is a statistical dependency index between two random variables.

Two ECE measures [14] that signify consistency of class labels w.r.t. clusters and have been used in data stream clustering are

$$\textbf{Purity}: \quad \frac{1}{n} \sum_{k=1}^{K} \max_{k^*} n_{k,k^*} \tag{6}$$

and

$$\textbf{Cluster-based cross entropy}: \quad \frac{1}{\log K^*} H(C^*|C). \tag{7}$$

Purity ranges from 0 to 1: the higher the value is, the purer the class labels in a cluster. Cluster-based cross entropy also ranges from 0 to 1: but the lower the value, the smaller the uncertainty of C^* given C. Purity increases and cross entropy decreases as classes in each cluster become more uniform. But they are insensitive to the distribution of clusters in each class, and the performance of both is likely to improve when the number of clusters increases. To the extremity, a "perfect" clustering can be achieved by assigning each distinct point to a unique cluster, though the clusters could be far off from the classes. To counteract this effect, we introduce and define

$$\textbf{Class-based cross entropy}: \quad \frac{1}{\log K} H(C|C^*), \tag{8}$$

which considers clustering consistency w.r.t. classes. Class-based cross entropy ranges from 0 to 1: the lower the value, the smaller the uncertainty of C given C^*.

ECE measures that consider consistency of both the cluster and the class labels w.r.t. each other are seeing increasing importance in clustering. We introduce them to measure the performance of data stream clustering specifically.

V**-Measure** [12] is the weighted Harmonic mean of

$$\textbf{Homogeneity}: \quad h = \begin{cases} 1 & \text{if } H(C) = 0 \\ 1 - \frac{H(C^*|C)}{H(C)} & \text{otherwise} \end{cases} \tag{9}$$

and

$$\textbf{Completeness}: \quad c = \begin{cases} 1 & \text{if } H(C^*) = 0 \\ 1 - \frac{H(C|C^*)}{H(C^*)} & \text{otherwise} \end{cases} \tag{10}$$

defined by

$$V\textbf{-Measure}: \quad \frac{(\beta+1)hc}{\beta h + c}, \tag{11}$$

where the weights of c and h are 1 and β, respectively, with $\beta > 0$. As h and c are in $[0,1]$ and the higher the better, V-Measure is always non-negative and maximized to 1 if a clustering is homogeneous ($h = 1$), with every cluster containing points of a single class, and complete ($c = 1$), with every point in a given class assigned to the same cluster.

VI, or variation of information, quantifies the information content difference between clusters and classes. It is defined by [10]

$$\textbf{VI}: \quad VI(C,C^*) = H(C) + H(C^*) - 2I(C,C^*). \tag{12}$$

VI is non-negative: the less the value, the less information content difference between C and C^* and hence the better the performance.

We summarize the seven ECE measures. Purity, cluster-based cross entropy, and homogeneity consider the consistency of class labels in each cluster. Class-based cross entropy and completeness consider the consistency of clusters in each class. V-Measure and VI combine both aspects.

ECE is designed for generic clustering algorithms. For a data stream clustering algorithm, we study how its ECE performance converges to its theoretical limit, i.e., the ECE performance when memory is not limited. This translates to investigating the ECE performance as a function of window size or memory limit, not previously reported for studying the performance of data stream clustering algorithms.

3 Partial 2nd-Order Cluster Representations

The most typical partial 2nd-order cluster representations used for the summary statistics $\mathscr{C}_k$ of cluster k in a d-dimensional space can be summarized as

$$\mathscr{C}_k = \left\{\mu_k, \Sigma_k = \text{diag}(\sigma_{k1}^2, \ldots, \sigma_{kd}^2), \pi_k\right\}, \tag{13}$$

where μ_k is a (weighted) mean vector for cluster k, corresponding to the 1st moments, Σ_k is a diagonal matrix with the diagonal line being (weighted) variances of each dimension for cluster k, corresponding to the 2nd moments, π_k is the total number (weight) of points in cluster k. This cluster representation highly compactly summarizes historical data using a non-oblique ellipsoid for each cluster.

HPStream [1] is a recent data stream clustering algorithm that uses a partial 2nd-order cluster representation for each cluster called fading cluster structure, with effective dimension reduction and merging strategies. It utilizes an iterative approach for incremental updating historical clusters, by assigning a newly arrived point to the closest cluster based on the Manhattan segmental distance. Each cluster has a limiting radius. A point is added to the nearest cluster if it lies inside the limiting radius of that cluster. Otherwise, a new singleton cluster is created for the point; an old cluster with the least recent updating is deleted.

4 Full 2nd-Order Cluster Representations

A predicament of the partial 2nd-order cluster representation is that it is not expressive enough for clusters in an oblique ellipsoidal shape. We extend it to the full 2nd-order representation by including the full covariance matrix for cluster k in $\mathscr{C}_k$:

$$\mathscr{C}_k = \{\mu_k, \Sigma_k, \pi_k\}, \tag{14}$$

where μ_k is a (weighted) mean vector, Σ_k is a (weighted) full covariance matrix corresponding to the 2nd cross moments, and π_k is the total number (weight) of points in cluster k. This representation is equivalent to the Gaussian mixture model (**GMM**), a statistically mature semi-parametric cluster analysis tool for modeling complex distributions. Geometrically, mean is the location of a cluster; covariance is an ellipsoidal approximation of the shape and orientation of a cluster. Sample mean and covariance together are sufficient statistics for a multivariate normal (**MVN**) distribution. GMM can be estimated from a static sample using the Expectation Maximization (**EM**) algorithm, which guarantees local convergence, but not readily applicable to data streams due to unavailable historical data. Thus we use EM only for estimating the GMM on each window of newly arrived data. By testing the statistical equality between a newly detected cluster and a historical one on their mean and covariance, using the 2nd cross moments, we determine whether to merge the two clusters. In the two tests below, we use the following notations. Let $x_1, \ldots, x_n \in R^d$ be a sample of size n for X with mean vector μ_x and covariance matrix Σ_x, and $y_1, \ldots, y_m \in R^d$ be a sample of size m for Y with mean vector μ_y and covariance matrix Σ_y.

Equality between two mean vectors – The Hotelling's T^2 statistic can determine whether μ_x statistically equals μ_y. The null hypothesis is $\mu_x = \mu_y$. T^2 is defined by [7]

$$T^2 = \frac{nm}{n+m}(\bar{x} - \bar{y})^\top S_{xy}^{-1}(\bar{x} - \bar{y}), \tag{15}$$

where $\bar{x}$ and $\bar{y}$ are respectively the sample means of X and Y, and the pooled sample covariance matrix is

$$S_{xy} = \frac{\sum_1^n (x_i - \bar{x})(x_i - \bar{x})^\top + \sum_1^m (y_i - \bar{y})(y_i - \bar{y})^\top}{n+m-2}. \quad (16)$$

By [7], if the two samples are independently drawn from two independent MVN distributions with the same mean and covariance, then

$$\frac{n+m-d-1}{d(n+m-2)} T^2 \quad (17)$$

has an F distribution with d numerator and $n+m-d-1$ denominator degrees of freedom.

Equality between two covariance matrices – We determine whether the covariance matrix Σ_x statistically equals a given covariance matrix Σ_0. The null hypothesis is $\Sigma_x = \Sigma_0$. We first transform the sample by

$$Y = L_0^{-1} X, \quad (18)$$

where L_0 is a lower triangular matrix obtained by the Cholesky decomposition $\Sigma_0 = L_0 L_0^\top$. Let Σ_y be the covariance matrix of Y. The new null hypothesis becomes testing $\Sigma_y = I$, where I is the d-dimensional identity matrix. The test can be performed using the W statistic defined by [8]

$$W = \frac{1}{d} tr[(S_y - I)^2] - \frac{d}{n}\left[\frac{1}{d} tr(S_y)\right]^2 + \frac{d}{n}, \quad (19)$$

where S_y is the sample covariance matrix of Y, $tr(\cdot)$ is the trace of a matrix, and n is the sample size. Under the null hypothesis that S_y is identity, the test statistic $\frac{nd}{2}W$ has an asymptotic χ^2 distribution with $d(d+1)/2$ degrees of freedom. Ledoit and Wolf have shown that the above asymptotic is true as both d and n go to ∞, known as (n,d)-consistent [8].

Updating historical clusters – If a newly arrived cluster k and a historical cluster j pass both the mean and covariance tests, we consider them statistically equivalent and merge them into a new one. By the definitions of mean and covariance, we can derive for the merged cluster

$$\mu = \frac{n\pi_j \mu_j + n_k \mu_k}{n\pi_j + n_k}, \quad (20)$$

$$\begin{aligned}\Sigma = &\frac{(n\pi_j - 1)\Sigma_j + (n_k - 1)\Sigma_k}{n\pi_j + n_k - 1} \\ &+ \frac{n\pi_j \mu_j \mu_j^\top + n_k \mu_k \mu_k^\top}{n\pi_j + n_k - 1} - \frac{n\pi_j + n_k}{n\pi_j + n_k - 1} \mu\mu^\top,\end{aligned} \quad (21)$$

and

$$\pi = \frac{n\pi_j + n_k \mu_k}{n + n_k}, \quad (22)$$

where $n\pi_j$ is the weighted number of historical points in cluster j, and n_k is the weighted number of points in the newly arrived cluster k.

5 Full 3rd- and 4th-Order Cluster Representations

In the streaming data model, points may not come randomly over time from each class. Those points from a same MVN class may break into pieces and fall into different windows of time. Since each isolated piece may have distinct means and covariance matrices, the merging strategies based on 2nd-order cluster representations may not recognize these pieces as being in the same class. To splice these isolated pieces back into the original MVN class, we add the 3rd and 4th cross moments, M^3 and M^4, respectively, to the cluster representation as follows:

$$\mathscr{C}_k = \left\{\mu_k, \Sigma_k, M_k^3, M_k^4, \pi_k\right\}. \quad (23)$$

The definitions of the 3rd and 4th cross moments of X are

$$M^3 : E(X^q X^r X^s) = \frac{1}{n}\sum_{i=1}^n X_i^q X_i^r X_i^s, \quad q,r,s \in \{1,\ldots,d\} \quad (24)$$

$$M^4 : E(X^q X^r X^s X^t) = \frac{1}{n}\sum_{i=1}^n X_i^q X_i^r X_i^s X_i^s, \; q,r,s,t \in \{1,\ldots,d\}, \quad (25)$$

where q, r, s and t are the dimension indices of X.

This higher-order cluster representation fundamentally expands the landscape beyond the 2nd-order paradigm, and provides a much wider playground for incremental cluster merging strategies. Using statistical tests for normality through skewness and kurtosis computable from the 3rd and 4th cross moments, one can discover new MVN clusters by combining cluster pairs with distinct 2nd-order statistics. We shall see below that the choice of both statistics lies in the fact that they can be decomposed into combinations of the cross moments of subsets of data, though their original definitions involve original data.

Characterizing the asymmetry of a probability distribution, the **multivariate skewness** is defined by [9]

$$b_{1,d} = \sum_{r,s,t=1}^d \sum_{r',s',t'=1}^d Z^{rr'} Z^{ss'} Z^{tt'} M_{1,1,1}^{r,s,t} M_{1,1,1}^{r',s',t'}, \quad (26)$$

where $Z^{rr'}$ is the element at the r-th row and the r'-th column of the inverse sample covariance matrix S^{-1}, and $M_{1,1,1}^{r,s,t}$ is the 3rd central cross moment which can be estimated from the sample by

$$\frac{1}{n}\sum_{i=1}^n (x_i^r - \bar{x}^r)(x_i^s - \bar{x}^s)(x_i^t - \bar{x}^t) \quad (27)$$

where x_i^r, x_i^s, x_i^t are the r,s,t-th dimensions of x_i, and $\bar{x}^r, \bar{x}^s, \bar{x}^t$ are the r,s,t-th dimensions of sample mean vector $\bar{x}$. The null hypothesis is that the sample comes from a normal distribution. Under the null hypothesis, the statistic

$$A = nb_{1,d}/6 \quad (28)$$

has an asymptotic χ^2 distribution with $d(d+1)(d+2)/6$ degrees of freedom [9]. For $d > 7$, $\sqrt{2A}$ approximately has a unit-variance normal distribution with mean $[d(d+1)(d+2)-3]/3$ [9]. Although computing $M_{1,1,1}^{r,s,t}$ by definition involves all historical data, the following equivalent form enables updating $M_{1,1,1}^{r,s,t}$ with only the newly arrived data:

$$\begin{aligned} M_{1,1,1}^{r,s,t} =& E(X^r X^s X^t) - E(X^t)E(X^r X^s) - E(X^s)E(X^r X^t) \\ & - E(X^r)E(X^s X^t) + 2E(X^r)E(X^s)E(X^t). \end{aligned} \tag{29}$$

Characterizing the peakedness of a probability distribution, the **multivariate kurtosis**, is given by [9]

$$b_{2,d} = E[(X-\mu)^\top \Sigma^{-1}(X-\mu)]^2, \tag{30}$$

with an equivalent form of

$$b_{2,d} = \sum_{q=1}^{d}\sum_{r=1}^{d}\sum_{s=1}^{d}\sum_{t=1}^{d} Z^{qr}Z^{st}M_{1,1,1,1}^{q,r,s,t}, \tag{31}$$

where $M_{1,1,1,1}^{q,r,s,t}$ is the 4-th central cross moment and can be estimated by

$$\frac{1}{n}\sum_{i=1}^{n}(x_i^q - \bar{x}^q)(x_i^r - \bar{x}^r)(x_i^s - \bar{x}^s)(x_i^t - \bar{x}^t). \tag{32}$$

Although computing $M_{1,1,1,1}^{q,r,s,t}$ by definition requires all historical data, it can be decomposed to combinations of cross moments as follows:

$$\begin{aligned} M_{1,1,1,1}^{q,r,s,t} = & E(X^q X^r X^s X^t) - E(X^t)E(X^q X^r X^s) - \\ & E(X^s)E(X^q X^r X^t) - E(X^r)E(X^q X^s X^t) - \\ & E(X^t)E(X^q X^r X^s) + E(X^s X^t)E(X^q X^r) + \\ & E(X^r X^t)E(X^q X^s) + E(X^r X^s)E(X^q X^t) + \\ & E(X^q X^t)E(X^r X^s) + E(X^q X^s)E(X^r X^t) + \\ & E(X^q X^r)E(X^s X^t) - 3E(X^q)E(X^r)E(X^s)E(X^t), \end{aligned} \tag{33}$$

which allows it to be updated exactly with only newly arrived data. Under the null hypothesis of normality,

$$\frac{b_{2,d} - [d(d+2)(n-1)/(n+1)]}{\sqrt{8d(d+2)/n}} \tag{34}$$

asymptotically follows the standard normal distribution $N(0,1)$ [9].

A historical cluster and a newly arrived cluster can be merged using the updating formula in Eq. (20) to (22), if their combination passes the multivariate normality tests.

6 Empirical Performance Comparison

Using the seven ECE measures on one synthetic and two real data streams, we compared the performance of six different cluster representation configurations:

- "`no merge`" – no merging is done between clusters detected from the newly arrived data with historical ones;
- "`2nd order diagonal covariance`" – $\mathscr{C}_k$ includes a weight, a mean and the diagonal line of a covariance matrix;
- "`HPStream`" – $\mathscr{C}_k$ is the same as in "`2nd order diagonal covariance`";
- "`2nd order`" – $\mathscr{C}_k$ includes a weight, a mean and a full covariance matrix;
- "`4th order`" – $\mathscr{C}_k$ includes a weight, a mean, a covariance matrix, a 3-D 3rd cross-moment matrix and a 4-D 4th cross-moment matrix; and
- "`4th order with one-dim`" – $\mathscr{C}_k$ is the same as in "`4th order`", but the merging strategy performs additional tests on the 1-D skewness and kurtosis.

We included lower-order cluster representations for comparison without the effect due to different merging strategies for the lower-order statistics. The performance is compared as a function of exponentially increasing window sizes. The EM algorithm we used was implemented in R package MClust [6]. The maximum number of clusters set for EM was 30. To correct the multiple simultaneous testing effect in comparing multiple pairs of clusters, we used Bonferroni adjusted p-values with an α-level of 0.05. The parameters for HPStream were set as follows: `InitNumber` = `speed` v = `window size`, `decay-rate` $\lambda = 0.5$, `spread radius factor` $\tau = 2$. We also set the `number of projected dimensions` to the original number of dimensions such that the results can be compared. All the experiments were performed on a Xeon 5135 CPU computer with 16GB memory running on SuSE Linux.

The synthetic data stream – Five data streams of 5,000 time points each were randomly generated from a 2-D five-component GMM as defined in Table 1. Figure 1 shows a

Table 1. Parameters of the 2-D GMM.

Component	Mean vector	Covariance matrix	
1	21.03224	0.6906479	2.043568
$\pi_1 = 0.0698$	-25.38627	2.043568	6.270815
2	-11.03656	0.5340183	-1.615203
$\pi_2 = 0.024$	-40.86163	-1.615203	9.669167
3	46.20645	0.853299	-2.514397
$\pi_3 = 0.0976$	-48.90667	-2.514397	21.18846
4	7.429518	0.7738911	0.3412344
$\pi_4 = 0.0988$	26.4398	0.3412344	7.950545
5	37.33823	0.04053236	-0.04607913
$\pi_5 = 0.7098$	-45.89366	-0.04607913	2.024866

sample of the data stream. Points belonging to different clusters are marked in distinct colors and symbols. The

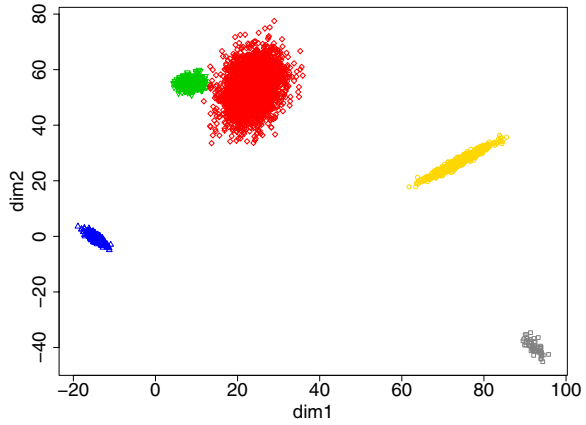


Figure 1. Scatter plot of the synthetic data stream.

proportions of each class evolve over time in a window-size step of 200, as shown in Table 2. The clustering performance was averaged over these five instances of data streams. The window sizes used in clustering were 157,

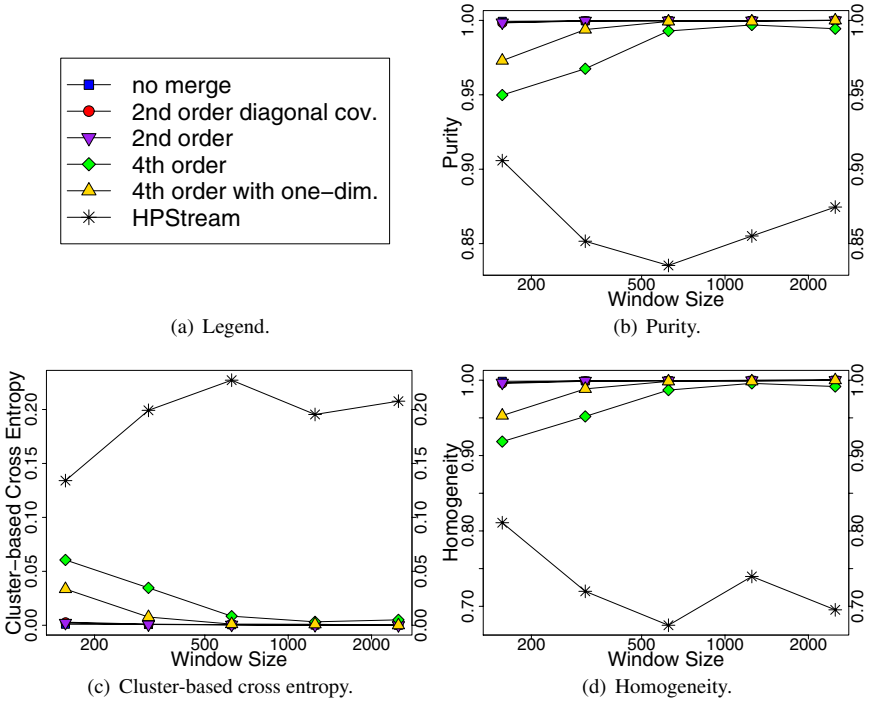
Table 2. Class proportions in each window.

Window	Class 1	Class 2	Class 3	Class 4	Class 5
1	0.062	0.152	0.452	0.000	0.334
2	0.064	0.058	0.194	0.000	0.684
3	0.074	0.030	0.112	0.000	0.784
4	0.050	0.000	0.076	0.000	0.874
5	0.094	0.000	0.044	0.008	0.854
6	0.070	0.000	0.050	0.104	0.776
7	0.074	0.000	0.034	0.164	0.728
8	0.066	0.000	0.010	0.212	0.712
9	0.070	0.000	0.004	0.244	0.682
10	0.074	0.000	0.000	0.256	0.670

313, 625, 1,250, and 2,500. The legend for different cluster representations is shown in Fig. 2(a). The performances using three types of ECE measures are shown in Fig. 2 to 4. The measures for consistency of classes given clusters are shown in Fig. 2; those measures for consistency of clusters given classes Fig. 3; and those combining both consistencies Fig. 4. We can observe that although the consistency of classes w.r.t. clusters is very similar (Fig. 2), there is a substantial difference in the consistency of clusters w.r.t. classes (Fig. 3): Higher-order cluster representations are much better than lower-order ones. The combination of both consistency measures (Fig. 4) leads to an overwhelmingly outperforming 4th-order cluster representation. The different ECE measures are also consistent. The under-performing lower-order cluster representations improve significantly over increasing window sizes, and eventually come close to the performance of the higher-order ones, because the various issues addressed by the higher-order cluster representations in data stream clustering become less dominant when the window size becomes larger. It may be worthwhile to point out that the performance of HPStream is very similar to the partial 2nd-order cluster representation, despite of different merging strategies.

The PDMC data stream – This data set was collected by BodyMedia, Inc. and used by the Physiological Data Modeling Contest (PDMC) in the International Conference on Machine Learning in 2004. Each point is described by sixteen attributes: userID, sessionID, sessionTime, two characteristics, annotation, gender and nine sensor values, with a class label of various physical activities. The training data set was collected by 9 sensors observing for approximately 10,000 hours, when a person wore a device. The objective was to cluster the points with several possible physical activity labels. We extracted the 9 sensory values from the data set containing a total of 46,209 observations. We used the window sizes of 723, 1,445, 2,889, 5,777, and 11,553 for performance evaluation. It took 1 hour and 44 minutes for the window size of 723. Other window sizes used less time. Figure 5 to 7 plot the performance on three types of ECE measures. Figure 5(a) shows the legend used for each picture. Figure 5 displays the performance evaluated by the measures which consider the consistency of classes w.r.t. clusters. Although the improvement by using higher-order strategies is not obvious, our merging strategies under various configurations always have better performance than HPStream. Figure 5(b) and Fig. 5(d) show that the purity and homogeneity of our method are always higher (better) than HPStream. Figure 5(c) shows that the cluster-based cross entropy of our method is always lower (better) than HPStream. Figure 6 shows the performance evaluated by the measures which consider the consistency of clusters in each class. In Fig. 6(a), although HPStream can achieve similar or even better performance than our method when the window size is small, its class-based cross entropy is always higher (worse) than our method when the window size is large enough. Figure 6(b) shows that the completeness of our method is always higher (better) than HPStream. Figure 7 displays the performance evaluated by the measures which consider both class and cluster consistencies. The V-Measure of our algorithm is always higher (better) than HPStream. The VI of our algorithm is always lower (better) than HPStream.

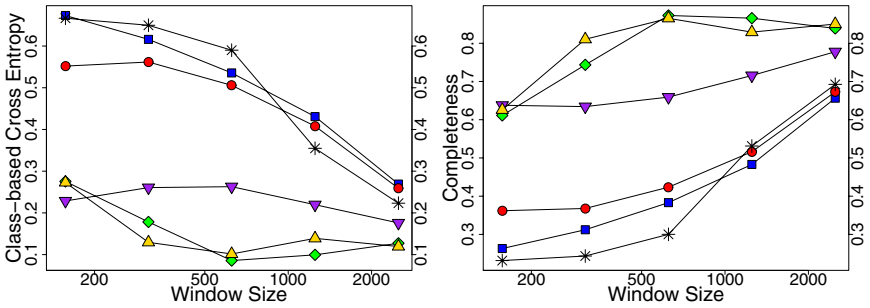
The CovType data stream – The forest CovType data set was obtained from the UCI KDD archive, used by several other papers for data stream clustering evaluation. There are a total of 581,012 instances in the data set and



(a) Legend. (b) Purity.

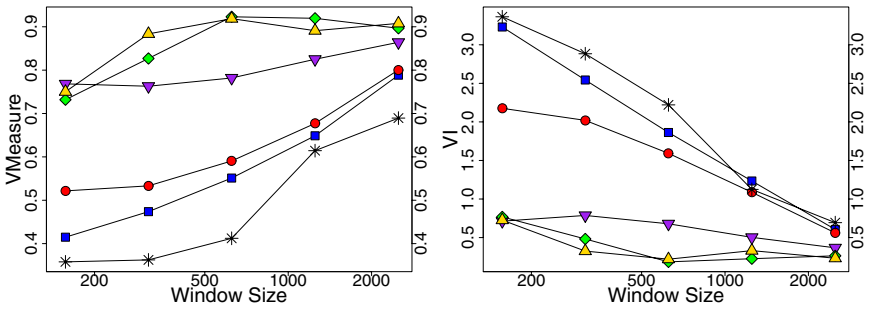
(c) Cluster-based cross entropy. (d) Homogeneity.

Figure 2. Cluster-based performance comparison on the synthetic data stream.



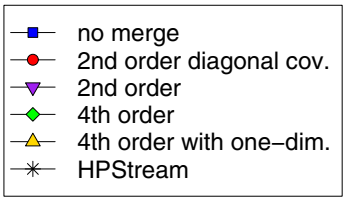
(a) Class-based cross entropy. (b) Completeness.

Figure 3. Class-based performance comparison on the synthetic data stream.

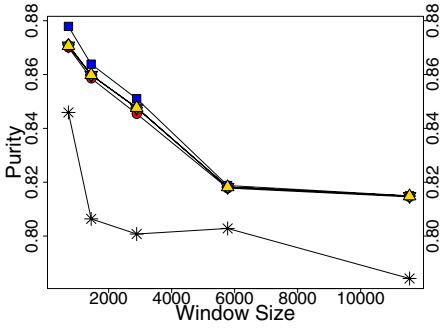


(a) V-Measure ($\beta = 1$). (b) VI.

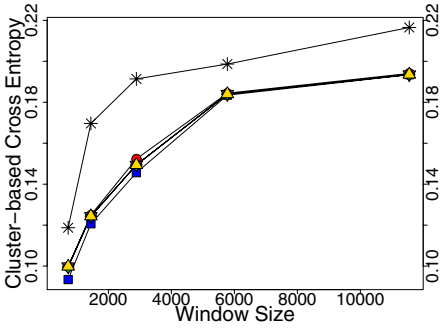
Figure 4. Cluster & class-based performance comparison on the synthetic data stream.



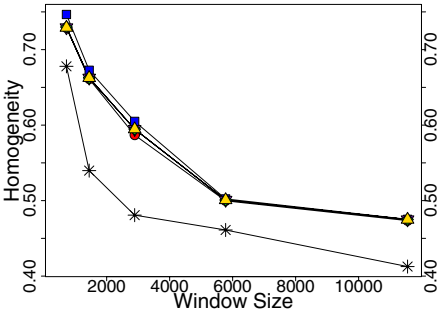
(a) Legend.



(b) Purity.

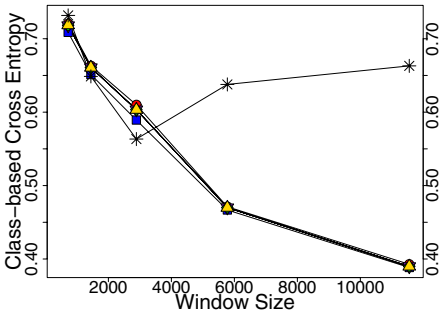


(c) Cluster-based cross entropy.

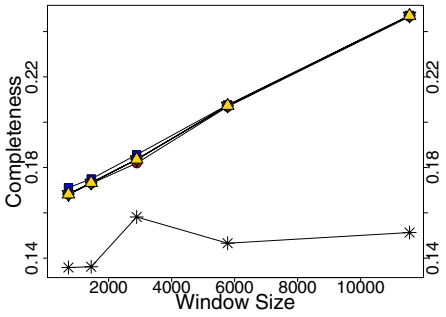


(d) Homogeneity.

Figure 5. Cluster-based performance comparison on the PDMC data stream.

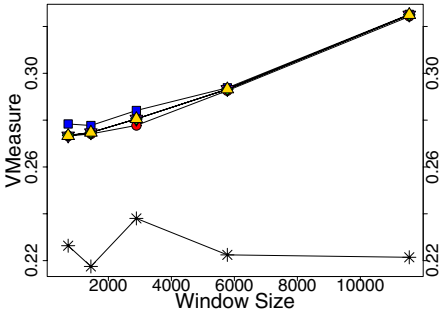


(a) Class-based cross entropy.

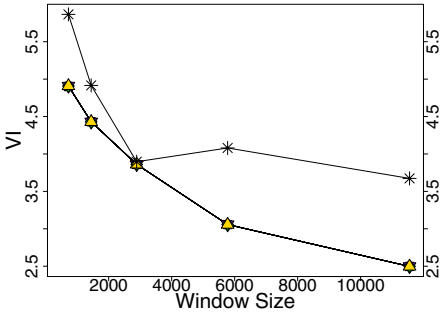


(b) Completeness.

Figure 6. Class-based performance comparison on the PDMC data stream.

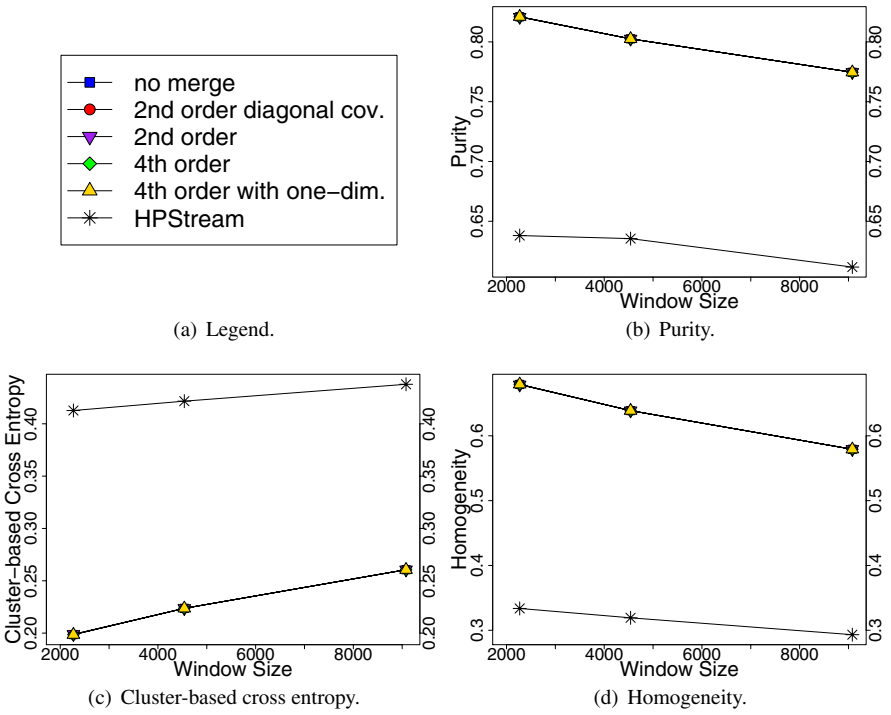


(a) V-Measure ($\beta = 1$).



(b) VI.

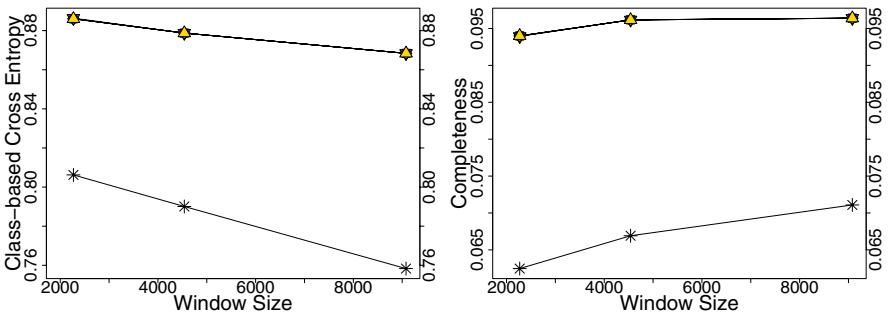
Figure 7. Cluster & class-based performance comparison on the PDMC data stream.



(a) Legend. (b) Purity.

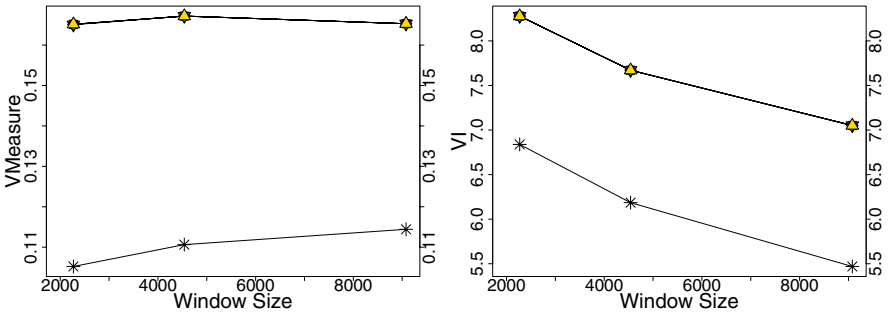
(c) Cluster-based cross entropy. (d) Homogeneity.

Figure 8. Cluster-based performance comparison on the CovType data stream.



(a) Class-based cross entropy. (b) Completeness.

Figure 9. Class-based performance comparison on the CovType data stream.



(a) V-Measure ($\beta = 1$). (b) VI.

Figure 10. Cluster- and class-based performance comparison on the CovType data stream.

seven different forest cover types as class labels designated for every instance. Each instance is described by 54 attributes, including 10 quantitative and 44 binary attributes. We extracted all 10 quantitative attributes from the whole data set. We used the window sizes of 2,270, 4,540, and 9,079. The window size of 2,270 consumed the longest time, 132 hours, to cluster this data stream. Figure 8 to 10 display the performance of using various cluster representations. Figure 8 displays the performance evaluated by the measures which consider the consistency of classes in each cluster. The improvement of our algorithm under higher-order cluster representations is not distinguishable from the lower-order ones. However, our algorithm under various configurations has better performance than HPStream in several scenarios. The purity (Fig. 8(b)) and homogeneity (Fig. 8(d)) of our method are always higher (better) than HPStream. The cluster-based cross entropy (Fig. 8(c)) of our algorithm also outperforms HPStream consistently. Figure 9 displays the performance evaluated by the measures which consider the consistency of clusters in each class. However, the performance by using these two methods – class-based cross entropy and completeness – is inconsistent: The class-based cross entropy (Fig. 9(a)) of our method is always higher (worse) than HPStream; the completeness (Fig. 9(b)) of our method is always above 0.09, better than HPStream (always below 0.07). Figure 10 plots the performance evaluated by those measures that combine consistencies w.r.t. both classes and clusters. The V-Measure (Fig. 10(a)) of our algorithm is always higher (better) than HPStream. But the VI (Fig.10(b)) of our algorithm is higher (worse) than HPStream, inconsistent with the V-Measure. Such inconsistency indicates the challenge of data stream clustering for which there is large room to improve the performance.

7 Conclusions

We have compared various cluster representations, from the partial 2nd order to full 4th order, of historical clusters for data stream clustering, on both synthetic and real data sets and compared them with the HPStream method. The seven ECE measures we employed consider not only the consistency of classes in each cluster but also the consistency of clusters in each class, the latter more or less overlooked in the literature. We conclude that higher-order cluster representations may help when the underlying distribution can be more or less approximated by the Gaussian mixture model, and our algorithm outperforms HPStream in most scenarios that we have tested. One remaining challenging issue is the poor performance on the completeness of clustering. This may be partly because the complex shape of a cluster does not capture well with the way the the 3rd and 4th cross moments are used in the higher-order cluster representation, though it is the most complex cluster description one can do so far. This opens up the problem of finding even more complex cluster representations for many real data streams. The higher-order cluster representation fundamentally expands the landscape beyond the 2nd-order paradigm, and provides a much wider playground for incremental cluster merging strategies. Additionally, multivariate kurtosis and skewness are only one avenue of utilizing the 3rd and 4th cross moments. One may expect to see other potentially powerful merging strategies based on the higher-order cross moments in the development of data stream clustering.

References

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proceedings of the VLDB Conference*, pages 81–92, 2003.

[2] J. Beringer and E. Hullermeier. Online clustering of parallel data streams. *Data and Knowledge Engineering*, 58:180–204, 2006.

[3] P. S. Bradley, U. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pages 9–15, 1998.

[4] F. Cao, M. Ester, W. Qian, and A. Zhou. Density-based clustering over an evolving data stream with noise. In *Proceedings of SIAM Conference on Data Mining*, 2006.

[5] B. E. Dom. An information-theoretic external cluster-validity measure. Research Report RJ 10219, IBM, 2001.

[6] C. Fraley and A. Raftery. MCLUST Version 3 for R: Normal Mixture Modeling and Model-Based Clustering. Technical Report no. 504, Department of Statistics, University of Washington, 2006.

[7] H. Hotelling. The generalization of Student's ratio. *Annals of Mathematical Statistics*, 2:360–378, 1931.

[8] O. Ledoit and M. Wolf. Some hypothesis tests for the covariance matrix when the dimension is large compared to the sample size. *Annals of Statistics*, 30(4):1081–1102, 2002.

[9] K. V. Mardia. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57:519–530, 1970.

[10] M. Meila. Comparing clusterings-an information based distance. *Journal of Multivariate Analysis*, 98:873–895, 2007.

[11] L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani. Streaming-data algorithms for high-quality clustering. In *Proceedings of the 18th International Conference on Data Engineering*, pages 685–694, 2002.

[12] A. Rosenberg and J. Hirschberg. V-Measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the Joint Conference EMNLP*, pages 410–420, 2007.

[13] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. *SIGMOD Record*, 25(2):103–114, 1996.

[14] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Techical Report, Department of Computer Science, University of Minnesota, 2001.