StreamSVC: A New Approach To Cluster Large And High-Dimensional Data Streams

Hasan Saberi, Mohammadali Mehdiaghaei

***Abstract*—The data stream mining has been studied extensively in recent years. This paper is introducing a novel method to cluster high-dimensional data streams, based on famous SVC method, named StreamSVC. SVC projects the images of the data points in a high–dimensional feature space, to search for the minimal enclosing sphere, then classifies the points with respect to the distance between each point's image and the central of feature sphere. In StreamSVC, for a single change in the data stream environment, the algorithm redoes the classification part. The algorithm involves only the parts of the data set which are affected during the change of stream and updates the classes in an appropriate time complexity order. Also, in order to update the clusters, in the stream process, we used some new improvements in the labeling piece of original SVC. These improvements are applied to reduce the computational costs for classification part and the cluster's labeling piece. The experimental results show both time efficiency and high accuracy for large data streams.**

***Index Terms*—Data stream, Clustering, SVC, Labeling piece.**

I. Introduction

THE process of grouping a set of data points into classes of similar data is called clustering. Lately, advancing on technology and communication systems, the data sets in stream form are widely generated. They are temporally ordered, fast changing, massive, and potentially infinite. It may be impossible to store an entire data stream or to scan through it multiple times due to its tremendous volume. The critical issues are Data Stream Management Systems and Stream Queries. Such queries and managements requires accurate and time efficient stream analysis. Clustering and cluster analysis are major ways to analysis the data sets. Thus clustering of high dimensional data stream is now a famous concept in mining of data streams.

There are some methodologies to deal with massive stream processing and stream systems. In order to handle these data types, some algorithm such as Random sampling, Sliding window or Histogram schemes are provided [1]. All of them get a part of total data set and reduces them into a abstracted data set. To obtain the clusters in a stream process, lots of researches have been performed [2], [3], [4], [5]. In the next section we introduce some of the new clustering approaches for data streams.

In (2001) Ben-Hur et al. [6] introduced SVC which is a kernel-based method for clustering massive and high–dimensional data sets. The algorithm applies a nonlinear projection to the data points to map their image into a high–dimensional data space, then searches for the minimal enclosing sphere in the feature space. After that, it classifies the points due to their distance from the central of feature sphere. There are two main bottlenecks here, first, pricy computation and second, poor labeling performance in the cluster's labeling piece. Recently many improvements are applied to solve the bottlenecks [7], [8], [9]. In this paper we applied a improved form of SVC and exchanged it into StreamSVC to achieve a strong and useful method for clustering the data streams.

The improved form of SVC we applied is using **SA** algorithm [8] to obtain an appropriate and time efficient algorithm.

In a quick review, StreamSVC applies SVC to initialize the first clusters, then updates the parameters related to SVC, cosequently updates the clusters. In an updating process, only some parts of the data set are affected. It is the strategy to reduce the time complexity order. As can be seen in the end of paper, the experimental results show high accuracy for large and high-dimensional data streams.

What we discuss in this paper: First we introduce SVC, then we discuss about the required issues about stream processing. At the third step we describe the lemmas which are the foundations of StreamSVC algorithm. Forth step is earmarked to the algorithm's pseudo-code, next to it, the experiential results are probed. The final step is conclusion.

II. Related works

O'Chalaghan et al. [2] proposed the STREAM algorithm to cluster data streams. STREAM is a k-means [10] based algorithm for clustering the data streams. The algorithm only makes a single pass over the data stream and uses small space. It requires $O(kN)$ time and $O(N^{\epsilon})$ space, where k is the number of centers, N is the length of data stream, and $\epsilon < 1$.

Aggarwal et al. [11] introduced a new approach to cluster the data streams called HPStream, a fading cluster structure, and the projection based clustering methodology. A fading cluster structure is a $(2d + 1)$ tuple, where d is number of dimensions, each tuple is an indicator of a micro-cluster. There are two main sections in algorithm (offline process and stream process). In the stream process, the algorithm calculate the *dimensions function* for $\overline{X}$ and the micro-clusters, then recisions the clusters and updates them. The offline section uses variant clustering approach to cluster the data set.

And also Aggarwal et al. [3] proposed CluStream algorithm. It adopts micro-clusters introduced in BIRCH algorithm [1] and uses micro-clusters to absorb arrived data points in online step. The offline step is to use k-means to cluster the micro-clusters into macro-clusters. The properties of micro-clusters are subtractive, so that according to the two snapshots of micro-clusters, clustering result can be got on every past

H. Saberi is with the Department of ComputerScience, ShahidBeheshti University Of Tehran, Iran, e-mail: (h_sabei_fie@yahoo.com).

M. Mehdiaghaei is with the Department of Computer Engineering, Azad University Of Tehran, Cental Branch, Iran, e-mail: (shabar_2k@yahoo.com).

time-horizon.
Tang et al. [12] introduced Movstream. The method focused on the cluster's shapes and the changes via definition of *Movement Event* include dieout, shrink, expand, and drift events, and operates on clusters which are the candidates to change.

III. SVC Method

We look for the smallest sphere in the Hilbert space that encloses the images of the data points [13], [14]. This sphere is mapped back to data space, where it forms a set of contours which enclose the data points. These contours are interpreted as cluster boundaries.

A. Description

Given a nonlinear transformation ϕ for a d-dimensional data point $X \in \mathcal{R}^d$ as $\phi(X)$, the distance between the transformed data point and the center of the sphere at the feature space is defined by:

$$\| \phi(X_j) - a \|^2 \leq R^2 + \xi_i, \; j = 1...N, \; \forall i, \xi_i \geq 0. \tag{1}$$

where $\| \; . \; \|$ is the Euclidean norm, R is the radius, a is the center of the feature sphere mapped by the data points and ξ_i are the slack variables. To solve the Eq.(1) we apply Lagrangian [14]:

$$L = R^2 - \sum_j (R^2 + \xi_j - \| \phi(X_j) - a \|^2)\beta_j \tag{2}$$
$$- \sum_j \xi_j \mu_j + C \sum_j \xi_j$$

where $\beta_j \geq 0$ and $\mu_j \geq 0$ are Lagrange multipliers, C is a constant, and $C \sum \xi_j$ is a penalty term. Setting to zero the derivative of L with respect to R, a and ξ_j, respectively, leads to

$$\sum_j \beta_j = 1 \tag{3}$$
$$a = \sum_j \beta_j \phi(X_j)$$
$$\beta_j = C - \mu_j \;\; so \;\; 0 \leq \beta_j \leq C,$$

for $j = 1...N$.
The KKT complementarity conditions of Fletcher [15] result in

$$\xi_i \mu_i = 0 \tag{4}$$
$$(R^2 + \xi_j - \| \phi - a \|^2)\beta_j = 0 \; .$$

To obtain the β_js, we eliminate the the variables R, a and μ_j , turning the Lagrangian into the Wolfe dual form that is a function of the variables β_j [14]

$$W = 1 - \sum_i \sum_j \beta_i \beta_j K(X_i, X_j), \tag{5}$$

where $K(a, b) = \exp(-q \parallel a - b \parallel^2)$. $K(a, b)$ is obtained from the inner product of the two ϕs $(\phi(a).\phi(b))$ in the Hilbert space where $\phi(a) = exp(-q \parallel x - a \parallel^2)$ [13]. Derivation with respect to β_j and considering the condition of Eq.(3) leads to

$$\beta_{n \times 1} = [A]^{-1}_{n \times n} B_{n \times 1}, \; \beta = [\beta_1 .. \beta_n]^T, \tag{6}$$

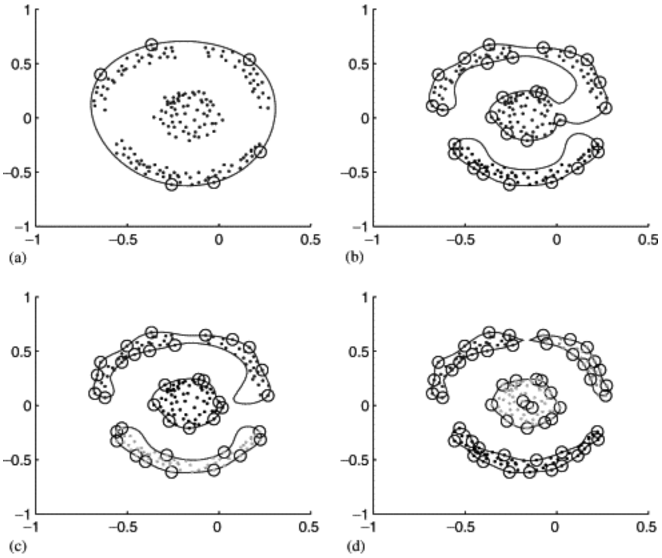

Fig. 1. Clustering of a data set containing 183 points using SVC with C =1. Support vectors are designated by small circles, and cluster assignments are represented by different gray scales of the data points. (a) q=1, (b) q=20, (c) q=24, (d) q=48.

where

$$A_{ij} = \begin{cases} 1, & i = 1 \\ -2K(X_i, X_j), & i \neq 1 \end{cases}, B_i = \begin{cases} 1, & i = 1 \\ 0, & i \neq 1 \end{cases}$$

At each point X, we define the distance of its image in feature space from the center of sphere as

$$R^2(X) = \| \phi(X) - a \|^2 \; . \tag{7}$$

In view of quadratic equation and the definition of the kernel [14], the following is got

$$R^2(X) = 1 - 2 \sum_j \beta_j K(X_j, X) \tag{8}$$
$$+ \sum_i \sum_j \beta_i \beta_j K(X_i, X_j).$$

The radius of the sphere is $R = \{R(x_j)\}$, x_j is support vector. The contours enclosed the points in data space are defined by the set $\{x | R(x) = R\}$.

B. Cluster Analysis

The number of **outlier** points are controlled by the parameter C. We have $N_{BSV} < 1/C$, where N_{BSV} is the number of Bounded Support Vectors (BSVs) or outliers. As $1/(CN)$ is an upper bound on the fraction of BSVs, thus $1/(CN) \in (0, 1]$. The value of the parameter C is related to the number of the data points and the willing, how much we want to avoid the outliers. The q of the $K(x, y)$ is width parameter of Gaussian kernel function. q and C influences the tightness and number of clusters and also the outlier points.

Fig.(1) shows an example of data points clustering with different qs without BSVs ($C = 1$). The contour of cluster is blur while q increases, and is fine while q decreases, but it makes the contour of the cluster affix mutually or break up if q is over-small or over-large. In Fig.(2a)without BSVs, contour separation does not occur for the two outer rings for any value of q. When some BSVs are present, the clusters are separated easily Fig.(2b). So the two parameters q and C are the identifier of the cluster's accuracy and tightness.

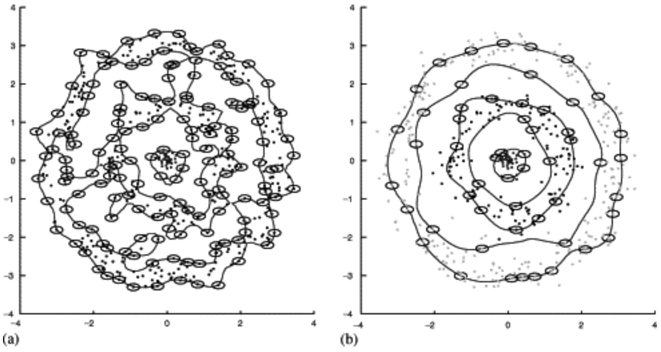

Fig. 2. Clustering with and without BSVs. The inner cluster is composed of 50 points generated from a Gaussian distribution. The two concentric rings contain 150/300 points, generated from a uniform angular distribution and radial Gaussian distribution. (a) The rings cannot be distinguished when C =1.0 Shown here is q=3.5, the lowest q value that leads to separation of the inner cluster. (b) Outliers allow easy clustering. The parameters are p=0.3 and q=1.0.

IV. Stream clustering

A. Stream processing

We consider the problem of clustering a data stream in the sliding window model [16]. The idea behind sliding window is to perform detailed analysis over the most recent data items and over summarized versions of the old ones. Consider n data points $X_{i_1}...X_{i_n}$ of a d-dimensional data space with time stamps $T_{i_1}...T_{i_n}$, the Sliding window (W_i), contains the last n income data points $X_{i_1}...X_{i_n}$, the new one overwrites on the oldest one (greatest time stamp) in the array memory of data points [17].

1) Updating βs: For each point X we can write the Eq.(8) as below

$$R^2(X) = \overline{R_k(X)} + R_k(X), \tag{9}$$

where

$$\overline{R_k(X)} = 1 - 2\sum_{j\neq k}\beta_j K(X, X_j) + \sum_{i\neq k}\sum_{j\neq k}\beta_i\beta_j K(X_i, X_j),$$

and

$$R_k(X) = 2\beta_k \sum_{i\neq k}\beta_i K(X_i, X_k) + \beta_k^2 + 2\beta_k K(X, X_k).$$

Consider the slide window W at time t_0, the points $X_{i_1}...X_{i_n}$, are clustered by SVC method as Fig.(2) with $\beta_1..\beta_n$ obtained by Eq.(6). If the oldest point X_k is overwritten by a new point X'_k at k^{th} location in memory array of data points, then for data points $X_{(i+1)_1}...X_{(i+1)_n}$ in W' we have

$$R'^2(X) = \overline{R'_k(X)} + R'_k(X). \tag{10}$$

where

$$\overline{R'_k(X)} = 1 - 2\sum_{j\neq k}\beta'_j K(X, X_j) + \sum_{i\neq k}\sum_{i\neq k}\beta'_i\beta'_j K(X_i, X_j),$$

and

$$R'_k(X) = 2\beta'_k \sum_{i\neq k}\beta'_i K(X_i, X'_k) + (\beta'_k)^2 + 2\beta'_k K(X, X'_k).$$

Remark 1: If n be an enough large number, β'_k can be obtained as follows

$$\beta'_k = \lambda\overline{\beta'_k} + (1-\lambda)\beta_z, \tag{11}$$

where

$$\overline{\beta'_k} = \frac{C\|X'_k - a_z\|^2}{\|v_* - a_*\|^2}.$$

v_* is a SV point in cluster $*$, with central point a_*, a_z is the nearest cluster center point and β_z is the lagrangian coefficient of nearest point to X'_k.
β' is the updated value of β caused by $X_k \rightarrow X'_k$ as $W \rightarrow W'$. If the recent change, causes $|P|$ changes in set β_js and $|p|$ changes in set $R^2(X_j)$s, j=1..n, then we can define the sets $P = \{j|\beta_j \neq \beta'_j\}$ and $p = \{j|R^2(X_j) \neq R^2(X_j)\}$.

Lemma 1: All β'_js, $j \in P$ can be obtained in $O(|P|^3)$, solving a linear system of form

$$\beta_{|P|\times 1} = [A]^{-1}_{|P|\times|P|} B_{|P|\times 1},\ \beta = [\beta'_{j\in P}]^T. \tag{12}$$

where

$$A_{i,j} = 2(\beta'_k K(X'_k, X_j) - K(X_i, X_j)),\ i \in \overline{p}. \tag{13}$$

Proof: We have $n - |p|$ values of $\Delta R^2(X_j) = 0$, where $\Delta R^2(X) = R'^2(X) - R^2(X)$. Using Eqs.(9, 10, 11), we can expand it as follows

$$\begin{aligned}\Delta R^2(X) = &-2\sum_{j\in P}(\beta'_j - \beta_j)K(X, X_j) \quad (14)\\ &+(\beta'_k)^2 - (\beta_k)^2 + 2(\beta'_k K(X'_k, X) - \beta_k K(X_k, X))\\ &+2\beta'_k \sum_{j\in P}(\beta'_j K(X_j, X'_k) - \beta_j K(X_j, X_k))\\ &+2\beta_k \sum_{j\in\overline{P}-\{k\}}\beta_j(K(X_j, X'_k) - K(X_j, X_k)).\end{aligned}$$

Using $|P|$ of $\Delta R^2(X)$s from set $\overline{p}$ and factoring the coefficients related to β's, we obtain the matrix A and B of linear system (12) and Eq.(13).

Remark 2: To obtain $|P|$ equations for linear system (12) we must have: $|P|, |p| < n/2$.
As the matrix A is not symmetric, obtaining A^{-1} is of order $O(|P|^3)$ [18].
Here, the critical issue is choosing an appropriate set P. Fig.(3) shows an example. As can be seen in the figure, the set P is composed by union of two sets, P_X (the circle with X as cental point) and $P_{X'}$ (the circle with X' as cental point). The updates are only happening around the deleted and added points, thus, we use the points in the two circles as sets P_X and $P_{X'}$. Setting a prefixed radius the circles can be obtained simply and finally the set $P = P_X \cup P_{X'}$.

2) Labeling piece: As mentioned before, labeling piece is a bottleneck in SVC. Ping et al. [8] introduced iSVC a novel approach, whose idea is to cluster the SVs firstly, then construct a classifier based on labeled SVs, finally label other data using the classifier. This algorithm is named as **SA** in some books. The steps are as follows

1) Create affinity matrix H with respect to SVs where H is a $V\times V$ matrix with $H_{i,j}$=$K(v_i, v_j)$. v_i and v_j are SVs.
2) Normalize H, using cholesky decomposition [19], into $H_c = D^{-1/2}HD^{-1/2}$, with $D_{ii} = \sum_j H_{i,j}$.

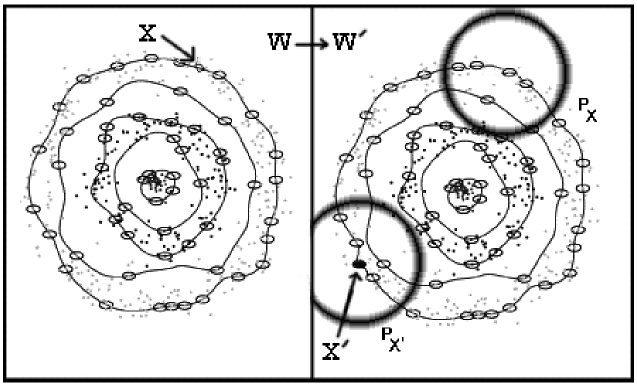

Fig. 3. As $X\rightarrow X'$, $W\rightarrow W'$, the data set is changed (including the neighbors of X and X'), consequently the shape of the related clusters.

3) Find $S_1..S_\kappa$, the κ largest eigenvectors (κ is specified by the number of eigenvalues that are larger than 1 [20]) and form Matrix $S_{V\times\kappa} = [S_{1_{V\times1}}...S_{\kappa_{V\times1}}]$ then normalize it: $\overline{S_{i,j}} = S_{i,j}^2/(\sum_j S_{i,j})^{1/2}$.
4) Treating each row of $\overline{S}$ as a point in $\mathcal{R}^\kappa$ and cluster it into κ clusters (using k-means [10]).
5) Label v_i as the i^{th} row's cluster membership.
6) Label other data in terms of its nearest SV's label.

Fig.(3) shows the changes of points position, with respect to previous clusters as $W\rightarrow W'$.
Consider the set $\delta = \{v_i' | i \in P \wedge v_i'\ is\ SV\}$. For data set P, we assume the matrix $H^\delta_{i,j}=K(v_i', v_j')$ with $v_i', v_j' \in \delta$ and apply above algorithm on it. After 6^{th} step, we have parted the dataset P, into its clusters.

Lemma 2: Cluster $\zeta^{P_{X'}}$ in P is a new cluster ζ' in W' if and only if

$$\forall\zeta\ in\ W : \zeta \cap \zeta^{P_{X'}} = \emptyset.$$

where ζs are the clusters of W and $\zeta^{P_{X'}}$ is a cluster in set $P_{X'}$.

Proof: Fig(4.a) shows a data set which contains one cluster and some outlier points. As $X\rightarrow X'$ we update the βs, then some new points are adding to SVs in the set P (the point in black circles are SVs). Applying **SA** algorithm on the Set $P_{X'}$, two clusters are obtained ($\zeta_1^{P_{X'}}$, $\zeta_2^{P_{X'}}$). As can be seen in Fig(4.b), One of these clusters ($\zeta_1^{P_{X'}}$) has no common point with the clusters in W, so it is assigned as new cluster in W'. In order to proof this lemma, this observation can be applied: Suppose the cluster $\zeta_1^{P_{X'}}$ can not be a new cluster in W', so it is either a part of cluster ζ in W or a mistaken output of **SA** algorithm. Trusting the **SA** algorithm, as the points in $\zeta_1^{P_{X'}}$ are outlier in W, clearly it is a new cluster in W'.

Lemma 3: Cluster $\zeta^P_{X'}$ in $P_{X'}$, unions with cluster ζ in W and resize it, and lead it to cluster ζ' in W' if and only if

$$(\zeta \cap \zeta^{P_{X'}} \neq \emptyset) \wedge (\forall\overline{\zeta}\ in\ W : \overline{\zeta} \cap \zeta^P = \emptyset),$$

where $\overline{\zeta}$ are the clusters of W except ζ.

Proof: In Fig(5.b), set P_PX' contains one cluster $\zeta_{P_{X'}}$. The same reasoning of previous lemma can be applied to proof. Because of $\zeta^{P_{X'}} \cap \overline{\zeta} = \emptyset$ the cluster ζ' can not be joined with $\overline{\zeta}$.

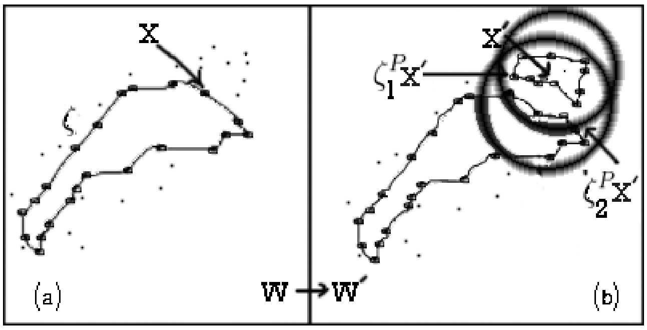

Fig. 4. As $X \rightarrow X'$, $W \rightarrow W'$, the data set is changed, adding X', some outlier points in W are now a new cluster in W'.

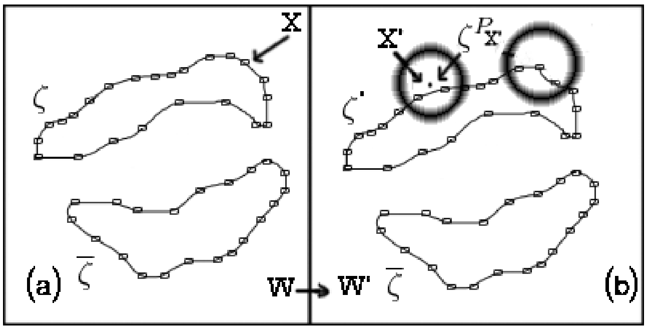

Fig. 5. As $X \rightarrow X'$, $W \rightarrow W'$, some reshaping in clusters are acquired in the cluster ζ.

Lemma 4: Cluster ζ in W, splits into clusters $\zeta_1'...\zeta_S'$ in W', if and only if there be S clusters $\zeta_1^{P_X}...\zeta_S^{P_X}$ in P_X, such that

$$(\zeta \cap \zeta_1^{P_X} \neq \emptyset) \wedge ... \wedge (\zeta \cap \zeta_S^{P_X} \neq \emptyset).$$

Proof: Fig(6.a) shows one cluster ζ in W. As can be seen in Fig.(6.b) the set P_X parts into two clusters $\zeta_1^{P_X}$ and $\zeta_2^{P_X}$. By same observation of previous lemmas, the cluster is splitting in W' environment.

Lemma 5: Clusters $\zeta_1...\zeta_M$ in W, merge into cluster ζ' in W', if and only if there be one cluster $\zeta^{P_{X'}}$ in $P_{X'}$, such that

$$(\zeta^{P_{X'}} \cap \zeta_1 \neq \emptyset) \wedge ... \wedge (\zeta^{P_{X'}} \cap \zeta_M \neq \emptyset).$$

Proof: Fig(7.a) shows two clusters ζ_1 and ζ_2. Adding X', set $P_{X'}$ (As can be seen in Fig(7.b)) contains one cluster, thus the two clusters are merged. Same as previous lemmas this observation can be applied: If the Clusters ζ_1 and ζ_2, are not merged by adding the new point X', then set $P_{X'}$, can not be contains one cluster. If so, the Algorithm would

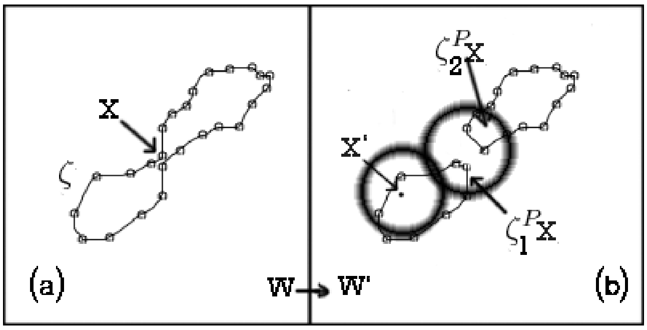

Fig. 6. As $X \rightarrow X'$, $W \rightarrow W'$, the cluster ζ splits into two clusters.

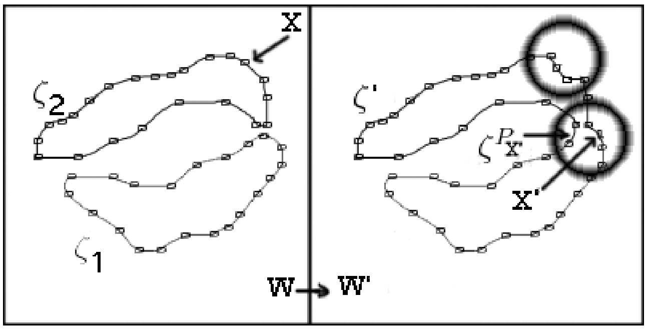

Fig. 7. As $X \to X'$, $W \to W'$, two clusters ζ_1 and ζ_2 are merged and composed on cluster.

be outputted a wrong cluster. Although the whole given proofs for the lemmas were intuitive, As the lemmas and the illustrations in examples are very clear, it is no needed for more details and more mathematical definitions.

B. StreamSVC Algorithm

The algorithm of StreamSVC method for a data set N, containing n data points of d-dimensional is as follows

(1) using SVC, for sliding window W, label the clusters as $\zeta_1 ... \zeta_\kappa$;
(2) as $X \to X'$, for new sliding window W':
(2 - 1) obtain the β' for X', using Eq.(11);
(2 - 2) compose the sets P and p (IV-A1, IV-A1);
(2 - 3) update βs for set P using lem.(1);
(2 - 4) update $\zeta_1 ... \zeta_\kappa$ to $\zeta'_1 ... \zeta'_{\kappa'}$:
(2 - 4 - 1) delete the clusters which has no BSVs;
(2 - 4 - 2) compose the new clusters using lem.(2);
(2 - 4 - 3) resize the clusters using lem.(3);
(2 - 4 - 4) split the clusters using lem.(4);
(2 - 4 - 5) merge the clusters using lem.(5);
(3) if the stream is not ending, goto step **(2)**;

The most important issue to increase the algorithm's accuracy is to compose the sets P and p (Step **2 - 2**). Because of the time complexity order of (Step **2 - 3**), to achieve overall complexity of $O(n)$, we must choose $|P| = \sqrt[3]{n}$. The appropriate points in set P and p, are the points, such that the points in $\overline{p}$ have relatively far distance from X and X'. We can suggest variant ways to achieve a good P and p. In the previous examples, simply a radius is prefixed and P is composed by union of P_X and $P_{X'}$ (see Fig.(3)). For the set p, we can simply choose $p = P$. Introducing the other ways to obtain P and p, we can observe following algorithm. Wang et al. [7] obtains a similarity matrix as follows: A link is created between a pair of points, r and s, if and only if r and s have each other in the list of their k_1 nearest neighbors, where k_1 is a user pre–specified parameter. The strength of a link between two points is expressed by the number of nearest neighbors that are shared by the two points, their similarity is defined as

$$sim(r, s) = |NN(r) \cap NN(s)|,$$

where $NN(r)$ and $NN(s)$ are the nearest neighbor list of r and s, respectively. Based on that we can introduce an algorithm as follows: obtaining the similarity matrix, we choose $|P|$ the number of points near X and X', the number of ones which have a strength similarity $(sim(X, X_i)) > \alpha)$, where α is a prefix number and composes set P_α.
Now choosing set p is quit simple. We just want the points in $\overline{p}$ which have relatively far distance from X and X', simply we can choose $p = P_\alpha$.

TABLE I
THE EXPERIMENTAL RESULTS OF IRIS DATA SET IN STREAM FORM. THE INITIAL PARAMETERS FOR SVC ARE $q = 4.2$, $C = 0.03$ AND FOR OBTAINING β'_k FROM EQ.(11), λ=0.5

Time	0.1	0.2	0.3	0.5	0.7	0.8	0.9
No. Clusters	3	3	3	3	3	3	3
Purity	0.97	0.95	0.95	0.94	0.93	0.93	0.93

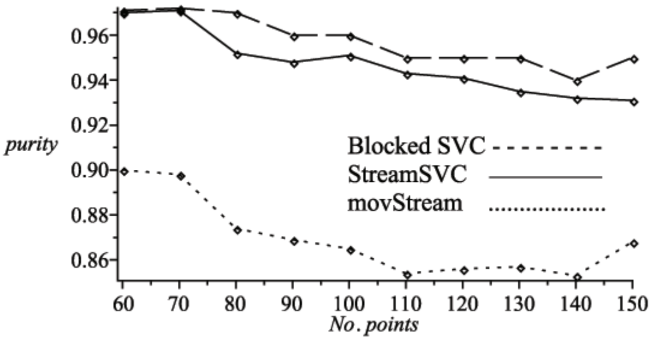

Fig. 8. Experimental results of iris data set based on purity, from 3 different methods. The Blocked SVC, StreamSVC, movStream with $|W| = 60$

V. EXPERIMENTAL RESULTS

This section evaluates the performance of our algorithm. The following experiments are conducted on Microsoft Windows XP Home Edition, with 1GB main memory and 2.4GHz CPU. The algorithm is tested with two different data sets. First one is iris data set [21] in stream form. The data set contains three clusters, and total 150 points (50 points for each cluster). We initialize the first sliding window by 60 points, 20 points from each cluster, then after each 0.01 second, periodically, a single point from each cluster is added.
Table I shows *purity* in timeline. Purity is average percentage of the dominant class label in each cluster [1]. In Fig.(7) the results are compared with movStream [12] (with *MaxNumCluster=3* and *MinNumCluster=2*) and Blocked SVC with initial values of q and C, same as Table I. Blocked SVC, reapplies the original SVC after every 10 points change. Because of the order of SVC, it cannot be an appropriating method for streams, but as it is good reference to examine the accuracy of StreamSVC, Thus, we compared it with our method. In this experience we choose set P as union of P_X and $P_{X'}$, where P_X is the set of point in a circle with X as central point and radius=1.

The second data set is KDD-CUP-99. The data set was created by Lincoln Labs, U.S.A. The data set contains a total of 24 attack types (connections) that fall into 4 major categories: Denial of service (Dos), Probe, User to Root (U2R), Remote to User (R2L). Each record is labeled either as normal, or as an attack, with exactly one specific attack

TABLE II
THE NUMBER OF EACH CASE IN THE PROBED SLIDING WINDOW.

W	10,001-11,000	80,001-81,000	100001-101000	210001-211000	310,030-311,029
sumrf	1000	0	0	345	0
snmpget	0	0	109	0	396
gsspass	0	8	10	0	0
nmap	0	0	1	0	0
portswp	0	0	12	0	0
satan	0	753	0	0	0
warezm	0	12	33	0	0

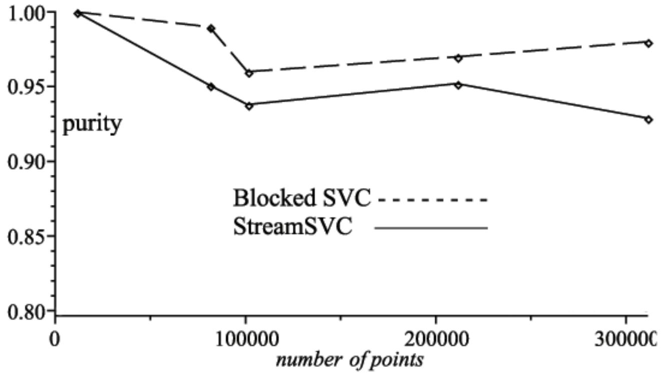

Fig. 9. Experimental results of KDD-CUP-99-corrected with Blocked SVC and SVC with the same C and q. For StreamSVC λ=0.65.

type. To examine the StreamSVC, we used a subset of whole data set named KDD-CUP-99-corrected. The dataset contains 311029 data points, and 42 dimensions. As in OCallaghan et al. (2002) and Aggarwal et al. (2003), all 34 continuous attributes will be used for clustering. We initialized the parameters as follows: $|W|$=1000, q=2.1, C=1 and the set P as union of two sets P_X and $P_{X'}$, where set P_X contains the 5 nearest point to X, so $|P|$=10. The initial sliding window, contains two kinds of attacks (186 cases of smurf and 103 cases of snmpgetattack) and the normal connections (711 cases), while from data point 210001 to 211000 we have 345 cases of smurf and 655 cases of normal connections. Table V, shows the number of each case in the probed W. Fig.(9) shows the purity of the clusters in the process and compared it with Blocked SVC. The result shows more than 93% accuracy for the method, and in the most areas, the deviation of its curve relatively to Blocked SVC's curve, is averagely 5-6%. The results shows that, the accuracy of the algorithm, is mostly equals to the original SVC algorithm for *non–stream* data sets. As we choose P=10, then total time is highly reduced, while the accuracy is still acceptable.

VI. CONCLUSION

In this paper we introduced StreamSVC, a novel algorithm to cluster high-dimensional data streams based on SVC method. StreamSVC applied SVC, to initialize the first clusters, then based on the changes in data environment, it takes a subsets of the dataset which are affected by the change, and reobtains the SVC's parameters to updates the clusters. The most critical part is preparing a good subset of the dataset which the complementary of it contains only the parts of dataset which surely are not changed in any of their parameters.

In the experimental result section, some real data sets are applied. The first one was the famous iris data set. As iris data set is a standard benchmark in the pattern recognition literature, we applied it in stream formatting. The reason was to assay the accuracy of the algorithm, and the second data set, KDD-CUP-99, to test both time efficiency and accuracy of clusters.

The experimental results shows high accuracy and time efficiently of the presented method. As the strength of original SVC is guaranteed, for every data sets, the accuracy and time complexity order can be acceptable.

ACKNOWLEDGMENT

With special thanks to Dr. L. Sharif and Prof. S. Kazem who helped the authors to execute the ideas and were great guiders through writing this paper.

REFERENCES

[1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2006, ch. 7, pp. 308–466.

[2] L. O'Callaghan, N. Mishra, S. G. A. Meyerson, and R. Motwani, "Streaming-data algorithms for high-quality clustering," *ICDE Conference*, 2002.

[3] C. Aggarwal, J. Han, J. Wang, and P. Yu, "A framework for clustering evolving data streams," in *In Proceedings of the International Conference on Very Large Data Bases (VLDB)*, 2003, pp. 81–92.

[4] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik, "Aurora: a new model and architecture for data stream management," *VLDB Journal*, vol. 12, no. 2, pp. 120–139, 2003.

[5] C. Aggarwal, *Data Streams: Models and Algorithms*. New York: Springer, 2007.

[6] A. BenHur, D. Horn, H. Siegelmann, and V. Vapnik, "Support vector clustering," *Machine Learning Research*, vol. 2, pp. 125–137, 2001.

[7] J.-S. Wang and J.-C. Chiang, "An efficient data preprocessing procedure for support vector clustering," *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 705–721, 2009.

[8] L. Ping, Z. Chun-Guang, and Z. Xu, "Improved support vector clustering," *Engineering Applications of Artificial Intelligence*, vol. 23, pp. 552–559, 2010.

[9] J. Lee and D. Lee, "An improved cluster labeling method for support vector clustering," in *Transactions on Pattern Analysis and Machine Intelligence*, ser. 3, vol. 27, 2005, pp. 461–464.

[10] R. Dua and P. Hart, *Pattern Classification and Scene Analysis*. J. Wiley and Sons, 1973.

[11] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "On high dimensional projected clustering of data streams," *Data Mining and Knowledge Discovery*, vol. 10, no. 3, pp. 251–273, 2005.

[12] L. Tang, C. J. Tang, L. Duan, C. Li, Y. X. Jiang, C. Q. Zeng, and J. Zhu, "Movstream: An efficient algorithm for monitoring clusters evolving in data streams," in *2008 IEEE International Conference on Granular Computing, GRC*, 2008, pp. 582–587.

[13] D. Horn, "Clustering via hilbert space," *Physica A*, vol. 302, pp. 70–79, 2001.

[14] D. Horn, A. BenHur, H. Siegelmann, and V. Vapnik, "A support vector method for clustering," in *Advances in Neural Information ProcessingSystems*, ser. Proceedings of the 2000 Conference, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13. MIT Press, 2001, pp. 367–373.

[15] R. Fletcher, *Practical Methods of Optimization*. Chichester: Wiley-Interscience, 1987.

[16] E. Ikonomovska, S. Loskovska, and D. Gjorgjevik, "A survey of stream data mining," in *Proceedings of 8th National Conference with International Participation, ETAI*, 2007, pp. 19–21.

[17] C. Aggarwal, H. J. wei, W. Jian-yong, and Y. P. S, "A framework for projected clustering of high dimensional data streams," in *Proceedings of the 30th International Conference on Very Large Data Bases, Toronto, Canada*, 2004, pp. 852–863.

[18] K. S. Miller, "On the inverse of the sum of matrices," vol. 54, no. 2, pp. 67–72, 1981.

[19] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York: Dover Publications, 1972.

[20] T. Zheng, L. Xiaobin, and J. Yanwei, "Disturbing analysis on spectrum clustering," *Science in China(SeriesE)*, vol. 37, no. 4, pp. 527–543, 2007.

[21] R. Fisher, "The use of multiple measurments in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.