Highly Efficient Incremental Estimation of Gaussian Mixture Models for Online Data Stream Clustering

Mingzhou Song[a,b] and Hongbin Wang[b]

[a]Department of Computer Science, Queens College of CUNY, Flushing, NY 11367, USA;
[b]Computer Science Ph.D. Program, Graduate Center of CUNY, New York, NY 10016, USA

ABSTRACT

We present a probability-density-based data stream clustering approach which requires only the newly arrived data, not the entire historical data, to be saved in memory. This approach incrementally updates the density estimate taking only the newly arrived data and the previously estimated density. The idea roots on a theorem of estimator updating and it works naturally with Gaussian mixture models. We implement it through the expectation maximization algorithm and a cluster merging strategy by multivariate statistical tests for equality of covariance and mean. Our approach is highly efficient in clustering voluminous *online* data streams when compared to the standard EM algorithm. We demonstrate the performance of our algorithm on clustering a simulated Gaussian mixture data stream and clustering real noisy spike signals extracted from neuronal recordings.

Keywords: Data stream clustering, Gaussian mixture models, expectation maximization, density merging

1. INTRODUCTION

The data stream clustering problem is defined as "to maintain a consistently good clustering of the sequence observed so far, using a small amount of memory and time".[1] The emphasis is limited time and memory relatively to the amount of data. Time-critical applications such as neuronal signal monitoring require realtime clustering. Memory-critical ones, for example, business transactions of a large retail company over the years, require massive data clustering with limited memory. Other applications are both time and memory critical.

We consider *recent data* as all the data available in the memory from the data stream. We define *historical data* as the data observed in the data stream so far, which include the recent data. Except for the portion of recent data, historical data are not available in memory. We call unprocessed recent data *newly arrived data*. If the entire historical data were available in memory, Gaussian mixture model would have been effectively estimated using the Expectation Maximization (EM) algorithm. For voluminous data streams, however, the EM algorithm is not efficient. For a data stream without complete historical records, the EM or any of its known variations is not applicable. We argue in this paper that we can adapt probability density based clustering algorithms to solve data stream clustering problems much more efficiently than applying the EM on the entire historical data. We apply the standard EM algorithm only on newly arrived data. Our incremental Gaussian mixture model estimation algorithm merges Gaussian components that are statistically equivalent. The equivalence of two Gaussian components are determined by the W statistic for equality of covariance and the Hotelling's T^2 statistic for equality of mean. The sufficient statistics of mean and covariance for a multivariate normal distribution make it possible to perform the tests and merging without resort to historical data.

Our idea is novel in the sense that we merge density components rather than data. With the strategy of merging data, the k-center problem has been addressed by efficiently maintaining clusters of small diameter as new points are inserted without keeping all historical data.[2] The authors propose a deterministic and randomized incremental clustering algorithm–the randomized doubling algorithm–with a provably good performance. It works in phases, each consisting of a merging and an updating stage. In the merging stage, the algorithm reduces the number of clusters by combining selected pairs; in the updating stage, the algorithm accepts new updates and tries to maintain at most k clusters without increasing the radius of the clusters or violating certain constraints.

Further author information: (Send correspondence to M.S.)
M.S.: E-mail: msong@cs.qc.edu, Telephone: 1 718 997 3584
H.W.: E-mail: hwang2@gc.cuny.edu

A phase ends when the number of clusters exceeds k. Guha *et al* give a constant-factor approximation algorithm for the k-median data stream clustering problem.[1] This algorithm makes a single pass over the data using small space. The medians are obtained hierarchically and clusters are built upon the intermediate medians. In contrast to the spatial clustering algorithms,[1,2] our clustering is density based. For density based clustering, the most effective method has been mixture models solved by the EM algorithm. The incremental EM and the lazy EM have been proposed to accelerating the EM using partial E-steps,[3] which are significantly more efficient for large and high dimensional databases. The former algorithm iterates through data blocks in a cyclic way. Each E-step updates only a part of conditional expectation for the complete data log-likelihood. For all other data blocks, previously computed log likelihood is reused. The initial pass needs all the data. The latter algorithm attempts to periodically identify significant cases and focuses on this subset of data. It has to keep all historical data in memory, which is the major limitation of both algorithms. Other papers have also reported on or are related to the data stream clustering problem.[4–9] But none of them has elaborated on an incremental estimation algorithm for Gaussian mixture models for online data stream clustering, which constitutes the topic of this paper.

Here we define some notations. Let T be the discrete time when the random data point X_T in $\mathbf{R}^d$ is observed. Here d represents the number of dimensions. We regard X_T as a random vector. Let $g^T : \mathbf{R}^d \to \mathbf{R}$ be an estimator of the true probability density function (p.d.f.) $p_0(x)$ based on the data points observed from the beginning time 1 to time T. We omit the superscript T when the value of T is not specified. Let $g^N(x)$ be an estimator of $p_0(x)$ based on the historical data $X_1, \cdots, X_N$. Let $g^{N+M}(x)$ be an estimator of $p_0(x)$ based on both the historical data $X_1, \cdots, X_N$ and the newly arrived data $X_{N+1}, \cdots, X_{N+M}$. The *data stream clustering problem* that we will address is: obtain $g^{N+M}(x)$ from $g^N(x)$ and the M newly arrived data sample $x_{N+1}, \cdots, x_{N+M}$. Here we assume that the cluster of each data point can be uniquely determined by $g(x)$.

We organize the content as follows. We give the theoretical foundation of our algorithm in Section 2. We introduce the algorithm for incrementally updating a GMM in Section 3. In Section 4, we illustrate the performance of the algorithm on clustering simulated Gaussian mixture data and clustering spike signals from neurons. Finally, we conclude our paper and point out further work in Section 5.

2. THE ESTIMATOR UPDATING THEOREM

Let $g(x)$ be $g^N(x)$. Let $q(x)$ be an estimator of $p_0(x)$ based on $X_{L+1}, \cdots, X_{N+M}$. $g(x)$ and $q(x)$ are very likely different but obviously related to each other because they share the data $X_{L+1}, \cdots, X_N$. Let $t(x)$ be an estimator of $p_0(x)$ based on $X_1, X_2, \cdots, X_L$, which are the data points removed from consideration of $q(x)$. Let $a(x)$ an estimator of $p_0(x)$ based on $X_{N+1}, X_{N+2}, \cdots, X_{N+M}$, which are the newly arrived data points included in $q(x)$. Under the assumption that the data points are independently and identically distributed (i.i.d.), we establish a theorem that dictates the way $q(x)$ is updated from $g(x)$ without exact knowledge of the overlapped data $X_{L+1}, \cdots, X_N$. When $L, M << N$, the theorem to be established will have direct impact on significantly improving the efficiency of the density estimation.

Theorem 1 (Estimator Updating Theorem). For the estimators $t(x), g(x), a(x)$ and $q(x)$ previously defined on data points in four time intervals within $[1, N+M]$, we have

$$q(x) = \frac{Ng(x) - Lt(x) + Ma(x)}{N - L + M} \tag{1}$$

This theorem is inspired by a relation among the empirical cumulative distribution functions (c.d.f.s). The empirical c.d.f.s $T(x), Q(x), G(x), A(x)$, corresponding to $t(x), q(x), g(x)$ and $a(x)$, respectively, are

$$\begin{aligned}
Q(x) &= \frac{|\{x_n \le x | n = L+1, L+2, \cdots, N+M\}|}{N - L + M} \\
G(x) &= \frac{|\{x_n \le x | n = 1, 2, \cdots, N\}|}{N} \\
T(x) &= \frac{|\{x_n \le x | n = 1, 2, \cdots, L\}|}{L} \\
A(x) &= \frac{|\{x_n \le x | n = N+1, N+2, \cdots, N+M\}|}{M}
\end{aligned}$$

From the four definitions above, we have evidently

$$LT(x) + (N - L + M)Q(x) = NG(x) + MA(x)$$

Thus, we obtain

$$Q(x) = \frac{NG(x) - LT(x) + MA(x)}{N - L + M}$$

If we could differentiate the above equation w.r.t. x, then we would get exactly Eq. (1). Unfortunately the differentiation can not be applied due to the non-smoothness of the empirical c.d.f. Here we have just offered an intuition behind the theorem claim. Now we prove the theorem using another strategy dealing directly with the p.d.f.s.

Proof. Let $f(x)$ be the estimator of $p_0(x)$ based on X_i for $i = 1, 2, \cdots, N + M$. We assume that the index i has a uniform distribution over $[1, N + M]$ with the probability mass function $P(i) = \frac{1}{N+M}$, for $i \in [1, N + M]$. By the definition of joint probability

$$f(x) = f(x|1 \leq i \leq N)P(1 \leq i \leq N) + f(x|N + 1 \leq i \leq N + M)P(N + 1 \leq i \leq N + M)$$

Since $g(x)$ and $a(x)$ are estimators based on $X_1, \cdots, X_N$ and $X_{N+1}, \cdots, X_{N+M}$ respectively, we have

$$f(x|1 \leq i \leq N) = g(x)$$

and

$$f(x|N + 1 \leq i \leq N + M) = a(x)$$

In addition,

$$P(1 \leq i \leq N) = \frac{N}{N + M}$$

and

$$P(N + 1 \leq i \leq N + M) = \frac{M}{N + M}$$

Thus we obtain

$$f(x) = g(x)\frac{N}{N + M} + a(x)\frac{M}{N + M} \tag{2}$$

Similarly, we can get

$$f(x) = t(x)\frac{L}{N + M} + q(x)\frac{N - L + M}{N + M} \tag{3}$$

Eliminating $f(x)$ from Eqs. (2) and (3), we find the original claim Eq. (1) true. □

Corollary 2 (Density of Entire History). For estimators $g^N(x)$, $g^{N+M}(x)$ and $a(x)$ previously defined on data points in the three time intervals within $[1, N + M]$, we have

$$g^{N+M}(x) = \frac{N}{N + M}g^N(x) + \frac{M}{N + M}a(x) \tag{4}$$

Proof. In Eq. (1), let $L = 0$, then $q(x)$ is identical to $g^{N+M}(x)$, the estimator of $p_0(x)$ based on the entire historical data so far. $g(x)$ is just another notation of $g^N(x)$. By the Estimator Updating Theorem, Eq. (4) follows immediately. □

3. UPDATING A GAUSSIAN MIXTURE MODEL WITH NEWLY ARRIVED DATA

Corollary 2 implies that in order to obtain an exact up-to-date density estimate, it is only necessary to use the newly arrived data without historical data. This fact is significant in that it will dramatically reduce the requirement for data storage and improve the efficiency of clustering. By Eq. (2), we can estimate $a(x)$ based on the newly arrived data first and then apply convex combination on $a(x)$ and $g^N(x)$ to produce $g^{N+M}(x)$. The drawback of this naïve application of Eq. (2) is that the estimated p.d.f. would not be compact and may be subject to overfitting to the data. The estimated p.d.f. by $g^T(x)$ will contain an increasing number of components as time goes on.

We adopt the Gaussian mixture model (GMM) for both $g(x)$ and $a(x)$ because the clustering of data points is straightforward with a GMM. A less obvious but crucial reason is that the sufficient statistics of a Gaussian component can be updated without using all historical data. The p.d.f. of a GMM is written as

$$\sum_{k=1}^{K} \pi_k \phi(x|\mu_k, \Sigma_k)$$

with

$$\sum_{k=1}^{K} \pi_k = 1,\ 0 \leq \pi_k \leq 1 \text{ for } k = 1, \cdots, K$$

where $\phi(x|\mu, \Sigma)$ is the p.d.f. of a multivariate normal distribution with mean vector μ and covariance matrix Σ:

$$\phi(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

We represent $g^N(x)$ by a GMM with parameters:

$$\pi_j, \mu_j, \Sigma_j,\ j = 1, \cdots, K_g$$

and $a(x)$ by a GMM with parameters:

$$\pi_k, \mu_k, \Sigma_k,\ k = 1, \cdots, K_a$$

where K_g and K_a are the numbers of components in each GMM respectively. To simplify notation, we use j to index components in $g^N(x)$ and k to components in $a(x)$, exclusively.

To maintain the compactness of a GMM and avoid overfitting, we design a strategy to merge statistically equivalent components into one. Here we do not employ any dependency among data blocks. We incrementally apply the EM algorithm on newly arrived data to produce the best estimator $g(x)$. We first obtain a GMM for $a(x)$ from the newly arrived data $x_{N+1}, \cdots, x_{N+M}$. Number of components K_a in $a(x)$ is selected using the Bayesian Information Criterion (BIC) defined by

$$-2 \log \mathcal{L}(x_{N+1}, x_{N+2}, \cdots, x_{N+M}) + \nu \log M$$

where $\mathcal{L}$ is the likelihood of $a(x)$ for the data and ν is the degrees of freedom of the parameters. For $a(x)$, a d-dimensional GMM with K_a components, ν is $[K_a(d+1)(d+2)/2] - 1$. According to the GMM of $a(x)$, we separate the newly arrived data into K_a clusters. Let $\mathcal{D}^k$ be the collection of data in cluster k. Let M_k be the number of data points in $\mathcal{D}^k$. For each cluster k, we will determine if it has a statistically equivalent covariance, using W statistic, and mean, using Hotelling's T^2 statistic, with any of the components in $g^N(x)$. We test mean equality after covariance equality because the Hotelling's T^2 test assumes equality of covariances. If any equivalent component j is found, then we create a new component in $g^{N+M}(x)$ by merging the component j of $g^N(x)$ with the component k of $a(x)$. If not, we will add the component k of $a(x)$ as a new component to $g^{N+M}(x)$ with an adjusted weight. All the remaining components in $g^N(x)$ will be added to $g^{N+M}(x)$ with weight adjusted accordingly. In the end, we merge statistically equivalent components in $g^{N+M}(x)$ with a similar strategy. The overall algorithm is shown as Alg. 1.

Algorithm 1 Incremental Gaussian Mixture Model Estimation

INPUT: GMM $g^N(x)$, newly arrived data $x_{N+1}, \cdots, x_{N+M}$
OUTPUT: GMM $g^{N+M}(x)$
Perform EM algorithm to estimate the Gaussian mixture model $a(x)$ from the new data $x_{N+1}, \cdots, x_{N+M}$, with number of components K_a determined by BIC
Assign each new data x_m to the most likely component according to the conditional probability

$$Prob(k|x_m),\ k = 1, \cdots, K_a$$

for each component k in $a(x)$ **do**
  Let $\mathcal{D}^k$ be the collection of all data in component k
  for component j with mean μ_j and covariance Σ_j in $g^N(x)$ **do**
    Calculate the W statistic to determine if $\mathcal{D}^k$ has equal covariance with Σ_j
    if $\mathcal{D}^k$ has passed the covariance test **then**
      Perform the Hotelling's T^2 test to determine if $\mathcal{D}^k$ has the same mean μ_j
      if $\mathcal{D}^k$ has passed the mean test **then**
        Consider components k in $a(x)$ and j in $g^N(x)$ identically distributed
        Compute the log likelihood of component j in $g^N(x)$ for $\mathcal{D}^k$ to break possible ties
      end if
    end if
  end for
end for
for each pair of equivalent components in $g^N(x)$ and $a(x)$ **do**
  Create a new component in $g^{N+M}(x)$ by merging the pair using Eqs. (6,7, 8)
end for
for each remaining component k in $a(x)$ **do**
  Assign this component to $g^{N+M}(x)$ with an updated weight using Eq. (9)
end for
for each remaining component j in $g^N(x)$ **do**
  Assign this component to $g^{N+M}(x)$ with an updated weight using Eq. (10)
end for
Merge statistically equivalent components in $g^{N+M}(x)$
return $g^{N+M}(x)$

3.1. Testing for Equality to a Covariance Matrix

In this test, we determine if the covariance matrix of a sample $x_1, x_2, \cdots, x_n \in \mathbf{R}^d$ of size n is statistically equal to a given covariance matrix Σ_0. The null hypothesis H_0 is $\Sigma_x = \Sigma_0$. The sample is assumed to be multivariate normal. We will also require Σ_0 to be positive definite. The data are first transformed by

$$y_i = L_0^{-1} x_i,\ i = 1, \cdots, n$$

where L_0 is a lower triangular matrix obtained by Cholesky decomposition of Σ_0, that is, $\Sigma_0 = L_0 L_0^\top$. An equivalent null hypothesis H_0' becomes $\Sigma_y = I$, where I is d-dimensional identity matrix. The W statistic is given by:[10]

$$W = \frac{1}{d} tr[(S_y - I)^2] - \frac{d}{n}\left[\frac{1}{d} tr(S_y)\right]^2 + \frac{d}{n} \tag{5}$$

where S_y is the sample covariance of y, and $tr(\cdot)$ is the trace of a matrix. Under the null hypothesis, the test statistic $\frac{nWd}{2}$ has an asymptotic χ^2 distribution with $d(d+1)/2$ degrees of freedom, that is,

$$\frac{nWd}{2} \sim \chi^2_{d(d+1)/2}$$

Ledoit and Wolf have shown that the above asymptotic is true as both d and n go to infinity,[10] known as (n, d)-consistent. The authors also performed Monte Carlo simulation to confirm the finite sample size behavior of the W statistic. The computation of W statistic does not involve the inverse of sample covariance matrix, implying that the test can still be performed when the sample covariance matrix is singular.

3.2. Testing for Equality to a Mean Vector

Once the covariance of the sample $x_1, x_2, \cdots, x_n \in \mathbf{R}^d$ is tested to be equal to a given matrix Σ_0, we determine whether the sample mean is equal to a given vector μ_0. The null hypothesis H_0 is $\mu = \mu_0$. Hotelling's T^2 test is a procedure for multivariate normal data, which is a natural extension to the univariate F-test. The T^2 statistic is defined by $n(\bar{x} - \mu_0)^\top S^{-1}(\bar{x} - \mu_0)$,[11] where S is the sample covariance. Under the null hypothesis, $\frac{n-d}{d(n-1)}T^2$ has F distribution with d numerator degrees of freedom and $n - d$ denominator degrees of freedom, that is,

$$\frac{n-d}{d(n-1)}T^2 \sim F_{d,n-d}$$

In the T^2 statistic, the inverse of sample covariance is necessary. In case of small sample size, we can use Σ_0 in the covariance test to replace S.

3.3. Merging or Creating Components

If $\mathcal{D}^k$ passed both covariance and mean tests for component j of $g^N(x)$, we consider component k of $a(x)$ has statistically equivalent distribution with component j of $g^N(x)$. We merge them to create a component in $g^{N+M}(x)$ with mean μ, covariance matrix Σ and weight π. By definitions of mean and covariance, we can derive:

$$\mu = \frac{N\pi_j\mu_j + M_k\mu_k}{N\pi_j + M_k} \tag{6}$$

$$\Sigma = \frac{N\pi_j\Sigma_j + M_k\Sigma_k}{N\pi_j + M_k} + \frac{N\pi_j\mu_j\mu_j^\top + M_k\mu_k\mu_k^\top}{N\pi_j + M_k} - \mu\mu^\top \tag{7}$$

$$\pi = \frac{N\pi_j + M_k}{N + M} \tag{8}$$

For component j of $g^N(x)$, the expected number of points in this cluster is $N\pi_j$. The sufficient statistics we need here are the sample mean vectors μ_j, μ_k, sample covariance matrices Σ_j, Σ_k, weight π_j, and sample sizes N, M, M_k. Therefore, we have completely avoided historical data without losing any information with regard to the GMM. In case that $\mathcal{D}^k$ has passed both covariance and mean tests for two or more different components in $g^N(x)$, we pick the component that has the maximum log likelihood for $\mathcal{D}^k$.

For each remaining component k in $a(x)$ that does not have a statistically equivalent component in $g^N(x)$, we create a new component in $g^{N+M}(x)$ with mean $\mu = \mu_k$ and covariance $\Sigma = \Sigma_k$, but with a weight of

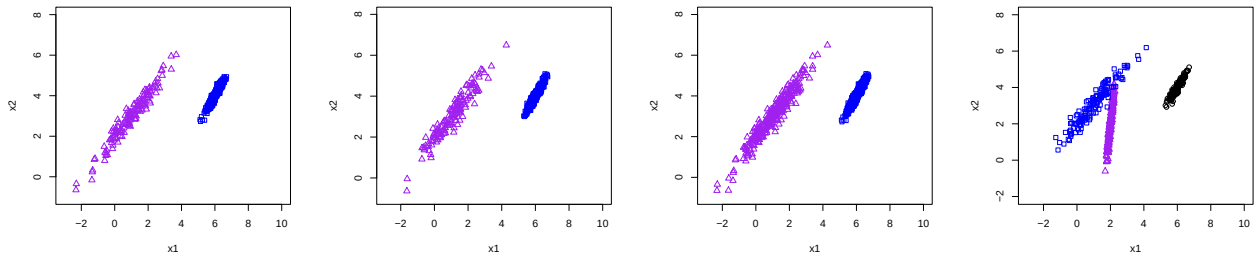
$$\pi = \frac{M_k}{N + M} \tag{9}$$

For each remaining component j in $g^N(x)$ that does not have a statistically equivalent component in $a(x)$, we create a new component in $g^{N+M}(x)$ with mean $\mu = \mu_j$ and covariance $\Sigma = \Sigma_j$, but with a weight of

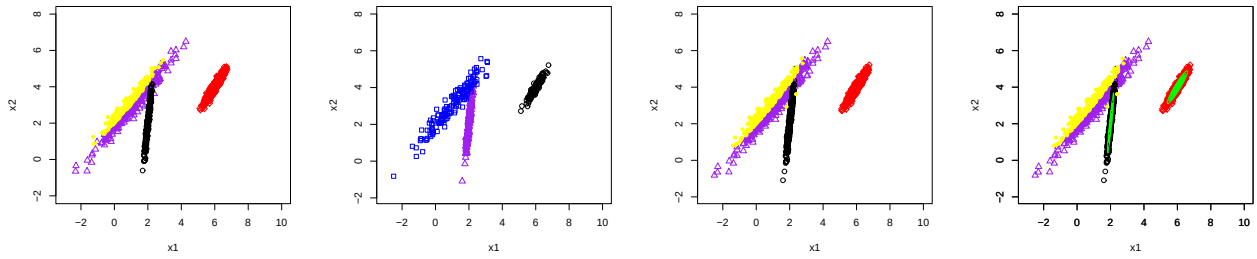
$$\pi = \frac{N\pi_j}{N + M} \tag{10}$$

4. EXPERIMENTAL RESULTS

We simulated data stream of 2000 points from a two-dimensional three-component GMM. The window size is 500. In the hypothesis testing for merging, the α-level is 0.02*. Figure 1 shows window by window clusters found by our algorithm: Figures 1 (a),(b),(d),(f) display clusters obtained using only data within each of the four windows; Figures 1(c),(e),(g) display clusters obtained after merging historical clusters with the current clusters. Figure 1(h) shows the final clusters with the density contour overlaid. The data points belonging to different clusters are marked by different symbols. From the pictures, we can tell that clustering within each of the four windows were all achieved correctly. Merging in Figs. 1(c) and (g) was also successful. However, the merging in Fig. 1(e) apparently failed because the yellow dots and purple triangles should really belong to the



(a) Clustering in 1st Window (b) Clustering in 2nd Window (c) Merging (a) and (b) (d) Clustering in 3rd Window



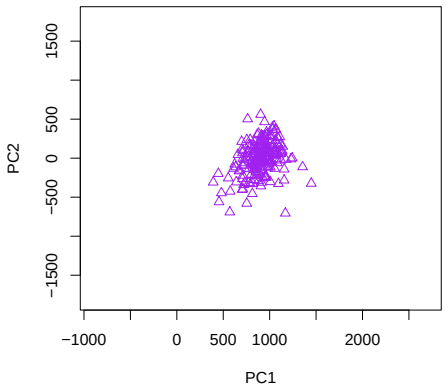
(e) Merging (c) and (d) (f) Clustering in 4th Window (g) Merging (e) and (f) (h) Final Clustering with Density Contour Plots

Figure 1. Clusters On Simulated Gaussian Mixture Data Stream With Density Contour Plots

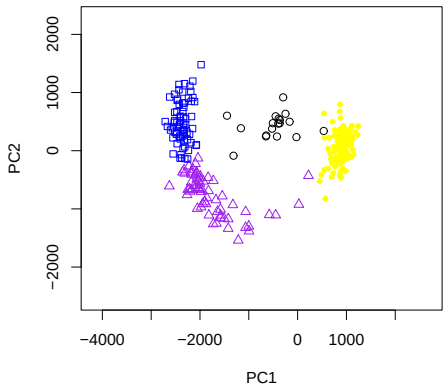
same cluster. This is an indication that further statistics beyond means and covariances for each cluster may be necessary to completely capture the information of each cluster.

A spike train is an extracellular action potential signal recorded with a probe, implanted in an animal subject under certain experimental condition. A spike train may be a superimposition of signals from several different neurons. An accurate model for a spike train is an additive mixture of serval time series. We assume no overlap between individual spikes. The spike train was separated into different windows by our preprocessing software implemented in C++. The candidate spikes were detected according to a low threshold in a spike window. The data points in each window formed a vector. There were 477 vectors in the data stream. Each vector had 200 dimensions. We reduced the dimension from 200 to 4 using principal component analysis. The window for newly arrived data contained 300 vectors. Figure 2 shows the clusters obtained by our algorithm: (a) shows the clusters obtained with the initial window; (b) shows the clusters obtained using the first window of newly arrived data; (c) shows the clusters after the final window. Figure 2(d) shows the solution by the standard EM algorithm applied on the entire data of the same spike train. In both figures, we only show the first two principal components. Figure 3 shows spike shapes of different clusters, obtained by our incremental algorithm, overlaid in one window. Each color indicates a different cluster corresponding to Fig. 2(c). Although for complex cluster shapes it differed with standard EM using all data, our algorithm produced a reasonable solution overall. The narrow long cluster was recognized as one cluster by the standard EM, but was broken into two by our incremental algorithm. The two clusters (purple triangle and the red diamond) in Fig. 2(c), which appeared to be the same and put in one cluster by standard EM in Fig. 2(d), were not merged by our incremental algorithm. From the spike shape plot in Fig. 3, we do observe certain differences between the black and the blue spikes and between the purple and the red spikes, suggesting that our clusters may catch some fine shape details.

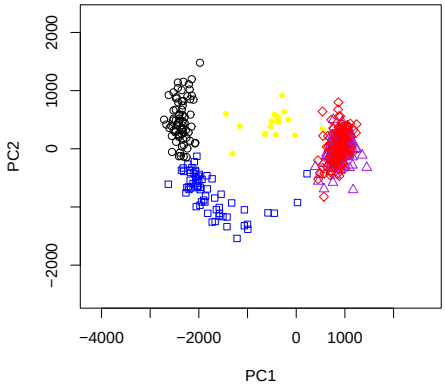
*We chose a α-level smaller than 0.05 to correct the multiple simultaneous test effect.



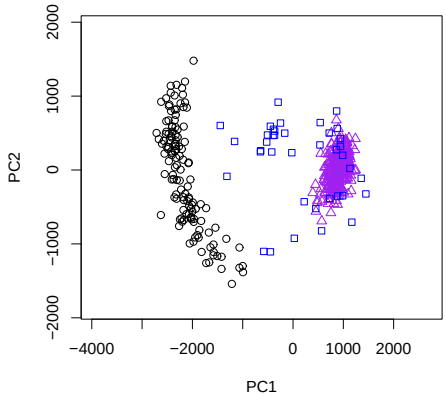
(a) Initial Clustering



(b) First Window



(c) Final Clustering



(d) Cluster Obtained with All Data by Standard EM

Figure 2. Clustering On Real Neural Spike Train Data Stream

5. CONCLUSION AND FUTURE WORK

We have presented an algorithm for online data stream clustering by merging components in GMM that are statistically equivalent. Instead of keeping all historical data, we have exploited the statistical structures of data streams using only newly arrived data. The experimental results have demonstrated that the algorithm is applicable to real world problems such as neuronal spike shape clustering.

Our algorithm does show a tendency to produce more clusters than the standard EM algorithm. Sometimes two clusters really belonging to one Gaussian component cannot be merged, because they do have different density individually. A better Gaussian component might have emerged had the two clusters been combined. We anticipate a collection of nearly sufficient statistics to be employed to improve the merging. We will address this issue in the next step.

Techniques directly improving the efficiency of the EM algorithm are fundamentally important and can be integrated into our framework.[3, 12]

The estimator updating theorem implies, in a more general setting, we may have to remove a component when

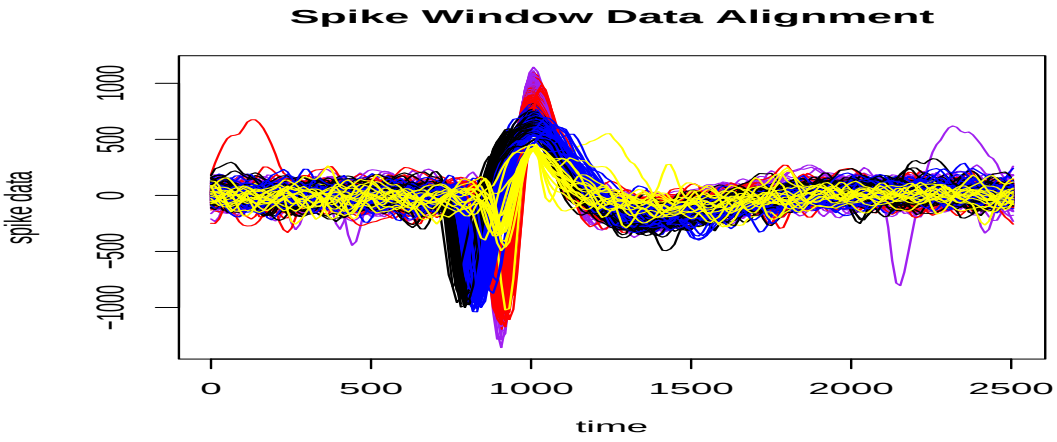


Figure 3. Overlay Of Spike Shapes In Different Clusters Obtained By The Incremental Algorithm

updating a density. It is not immediately obvious how the removal can be achieved, which we may investigate further.

Points in a data stream are usually dependent. Models that consider this dependency such as Markov models and regression models may also improve the performance of our algorithm.[13]

ACKNOWLEDGMENTS

The authors thank Plexon Inc. at Dallas, TX for providing the neuronal spike train data set.

REFERENCES

1. S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams," in *IEEE Symposium on Foundations of Computer Science*, pp. 359–366, 2000.
2. M. Charikar, C. Chekuri, T. Feder, and R. Motwani, "Incremental clustering and dynamic information retrieval," in *Proc. 29th Symposium on Theory of Computing*, pp. 626–635, 1997.
3. B. Thiesson, C. Meek, and D. Heckerman, "Accelerating EM for large databases," Tech. Rep. MSR-TR-99-31, Microsoft Research, Redmond, WA, May 1999. Revised Feb. 2001.
4. C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proc. 29th Int'l Conf. on Very Large Data Bases*, Sept. 2003.
5. P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 71–80, 2000.
6. V. Ganti, J. Gehrke, and R. Ramakrishnan, "Mining data streams under block evolution," *SIGKDD Explor. Newsl.* **3**(2), pp. 1–10, 2002.
7. G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proc. 7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 97–106, 2001.
8. G. Manku and R. Motwani, "Approximate frequency counts over data streams," in *Proc. 28th Int'l Conf. on Very Large Data Bases*, 2002.
9. G. J. McLachlan and D. Peel, "Robust cluster analysis via mixtures of multivariate t-distributions," in *Lecture Notes in Computer Science Vol. 1451*, pp. 658–666, Springer-Verlag, 1998.
10. O. Ledoit and M. Wolf, "Some hypothesis tests for the covariance matrix when the dimension is large compared to the sample size," *The Annals of Statistics* **30**(4), pp. 1081–1102, 2002.
11. H. Hotelling, "The generalization of Student's ratio," *Annals of Mathematical Statistics* **2**, pp. 360–378, 1931.

12. J. Q. Li and A. R. Barron, "Mixture density estimation," in *Advances in Neural Information Processing Systems 12*, S. A. Solla, T. K. Leen, and K.-R. M eds., MIT Press, 2000.
13. K. Fokianos and B. Kedem, "Regression theory for categorical time series," *Statistical Science* **18**, pp. 357–376, Aug. 2003.