Efficiently and Accurately Comparing Real-valued Data Streams
(Extended Abstract)

Paolo Capitani and Paolo Ciaccia

DEIS - IEIIT-BO/CNR, University of Bologna, Italy
{pcapitani,pciaccia}@deis.unibo.it

Abstract. Data streams are pervasive in many modern applications, and there is a pressing need to develop techniques for their efficient management. In this paper we consider real-valued streams and deal with the problem of reporting in real-time all the instants in which their distance falls below a given threshold. Current distance measures, such as Euclidean and Dynamic Time Warping (DTW), either are inaccurate or are too time-consuming to be applied in a streaming environment. We propose $SDTW$, a novel DTW-like distance measure which can be continuously updated in constant time and experimentally show that it improves over DTW by orders of magnitude without sacrificing accuracy.

1 Introduction

Management of data streams has recently emerged as one of the most challenging extensions of database technology. The proliferation of sensor networks as well as the availability of massive amounts of streaming data related to telecommunication traffic monitoring, web-click logs, geophysical measurements and many others, has motivated the investigation of new methods for their modelling, storage, and querying. In particular, continuously monitoring through time the correlation of multiple data streams is of interest, among others, in financial, video surveillance and biological applications, and, more in general, for mining temporal patterns.

Previous works dealing with the problem of detecting when two or more streams exhibit a high correlation in a certain time interval have tried to extend techniques developed for (*static*) time series to the streaming environment. In particular, Zhu and Shasha [7], by adopting a *sliding window* model and the Euclidean distance as a measure of correlation (low distance = high correlation), have been able to monitor in real-time up to 10,000 streams on a PC. However, it is known that for time-varying data a much better accuracy can be obtained if one uses the *Dynamic Time Warping* (DTW) distance [2]. Since the DTW can compensate for stretches along the temporal axis, it provides a way to optimally align time series that matches user's intuition of similarity much better than Euclidean distance does, as demonstrated several times (see, e.g., [5] for some recent DTW applications and [1] for a novel DTW-based approach to shape matching). Further, although not a metric, *DTW* can be indexed [3], which allows this distance to be applied also in the case of large time series archives.

Unfortunately, when considering streams the benefits of *DTW* seem to vanish since, unlike Euclidean distance, *it cannot be efficiently updated*. The basic reason is that the (optimal) alignment one has established at time t is not guaranteed to be still optimal at time $t+1$, thus forcing the *DTW* to be recomputed from scratch at each time step. Given this unpleasant state of things, in this paper we propose a novel *DTW-like* distance measure, called *SDTW* (*Stream-DTW*), which is efficiently updatable and it is a very good approximation of *DTW*.

We demonstrate that *SDTW* is orders of magnitude faster than *DTW* and that its accuracy is much better that currently known approximation techniques of *DTW* developed for the static case.

2 Dynamic Time Warping

We start with some basic definitions related to the static case, i.e., for real-valued time series of finite length. Let $R \equiv R_1^n$ and $S \equiv S_1^n$ be two time series of length n and let R_i (S_i) be the i-th sample of R (resp. S). The Euclidean (L_2) distance between R and S, $L_2(R_1^n, S_1^n) = \sqrt{\sum_{i=1}^{n}(R_i - S_i)^2}$, only compares corresponding samples, thus it does not allow for *stretches* along the temporal axis. As a consequence, two time series might lead to a high L_2 value even if they are very similar. This problem is solved by the *Dynamic Time Warping* (*DTW*) distance. The key idea of *DTW* is that any point of a series can be (forward and/or backward) aligned with multiple points of the other series that lie in different temporal positions, so as to compensate for temporal stretches.

Let d be the $n \times n$ matrix of pairwise squared distances between samples of R and S, $d[i,j] = (R_i - S_j)^2$. A *warping path* $W = \langle w_1, w_2, \ldots, w_K \rangle$ is a sequence of K ($n \leq K \leq 2n-1$) matrix cells, $w_k = [i_k, j_k]$ ($1 \leq i_k, j_k \leq n$), such that:

boundary conditions: $w_1 = [1,1]$ and $w_K = [n,n]$, i.e., W starts in the lower-left cell and ends in the upper-right cell;

continuity: given $w_{k-1} = [i_{k-1}, j_{k-1}]$ and $w_k = [i_k, j_k]$, then $i_k - i_{k-1} \leq 1$ and $j_k - j_{k-1} \leq 1$. This ensures that the cells of the warping path are adjacent;

monotonicity: given $w_{k-1} = [i_{k-1}, j_{k-1}]$ and $w_k = [i_k, j_k]$, then $i_k - i_{k-1} \geq 0$ and $j_k - j_{k-1} \geq 0$, with at least one strict inequality. This forces W to progress over time.

Any warping path W defines an alignment between R and S and, consequently, a cost to align the two series. The (quadratic) *DTW* distance is the minimum of such costs, i.e., the cost of the *optimal warping path*, W_{opt}:

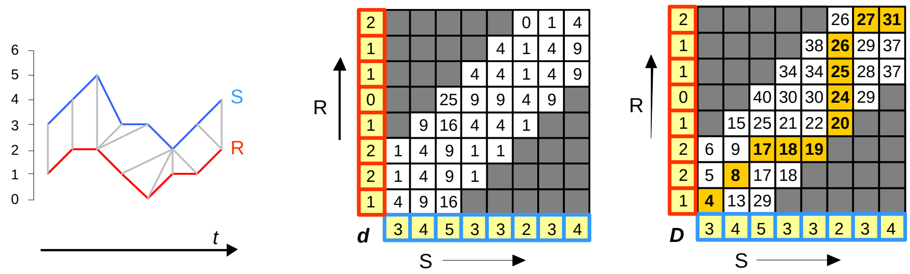
$$DTW(R_1^n, S_1^n) = \min_W \{ \sum_{[i_k,j_k] \in W} d[i_k, j_k] \} = \sum_{[i_k,j_k] \in W_{opt}} d[i_k, j_k] \qquad (1)$$

The *DTW* distance can be recursively computed using an $O(n^2)$ dynamic programming approach that fills the cells of a *cumulative distance matrix* D using the following recurrence relation (see also Figure 1):

$$D[i,j] = d[i,j] + \min\{D[i-1,j-1], D[i-1,j], D[i,j-1]\} \quad (1 \le i,j \le n)$$

and then setting $DTW(R_1^n, S_1^n) = D[n,n]$. Note that $D[i,j] = DTW(R_1^i, S_1^j)$, $\forall i,j$, since the DTW distance is also defined for series of different length.

In practical applications, warping paths are commonly subject to some *global constraint*, in order to prevent pathological alignments. The most commonly used of such constraints is the Sakoe-Chiba band [6] of width b ($b \in [0, n-2]$ integer), that forces warping paths to deviate no more than b steps from the matrix diagonal (see Figure 1 (b)). It is worth noting that, besides reducing the time and space complexity of DTW computation to $O(nb)$,[1] rather surprisingly a band of limited width b leads to better results in classification tasks [5].



(a) (b) (c)

Fig. 1. Computing the DTW distance with a Sakoe-Chiba band of width $b = 2$: (a) Optimal alignments; (b) The d matrix of pairwise sample distances; (c) The D cumulative distance matrix; highlighted cells constitute the optimal warping path W_{opt}

3 The Stream-DTW Distance

Let us go back to the streaming environment, in which R and S are now possibly infinite sequences of samples. We assume that at each time step a new sample for both streams becomes available (i.e., streams are synchronized). We consider the well-known *sliding window* model, according to which, given a sliding window of size n, only the last n points of the streams are significative. Thus, at time t, the subsequence of stream R falling in the current (active) window is R_{t-n+1}^t. At time $t+1$ sample R_{t+1} arrives and R_{t-n+1} expires, thus the new active window consists of R_{t-n+2}^{t+1}. The same is true for stream S.

If one wants to monitor the streams so as to report when their distance $dist$ falls below a given threshold value ϵ, a näive approach would be to compute, at time $t+1$, $dist(R_{t-n+2}^{t+1}, S_{t-n+2}^{t+1})$ from scratch, i.e., without considering the computation performed at previous time steps. This might be a challenging task, especially with high sampling rates, large sliding windows, and many (hundreds, thousands) streams. As anticipated in the Introduction, the DTW distance is not

[1] More precisely, the number of cells reduces to $n + 2\sum_{i=1}^{b}(n-i) = (2b+1)n - b(b+1)$.

efficiently updatable, since, in the general case, the optimal alignment established at time t is of no help to find the optimal alignment at time $t+1$. The resulting update cost is therefore $O(nb)$, which is clearly unsuitable especially with large sliding windows.

Given this unpleasant state of things, we move to consider a more realistic objective, that is, to devise a *DTW-like* distance measure suitable for streaming environments. The measure should fulfill the following requirements:

1. It should be fast to update, i.e., its complexity should be $O(b)$.[2] In particular, update complexity should be independent of the sliding window size, n.
2. It should be a good approximation of DTW. This is because of the success DTW has already demonstrated for the static case (i.e., time series).
3. It should be a *lower bound* of DTW, i.e., it should never overestimate the DTW value.

Let us briefly comment on the 3rd requirement. Although not strictly necessary, if our new distance measure, call it $SDTW$, is a lower bound of DTW, then, whenever we have $SDTW(R_{t-n+1}^t, S_{t-n+1}^t) \geq \epsilon$ we can also immediately conclude that $DTW(R_{t-n+1}^t, S_{t-n+1}^t) \geq \epsilon$, without performing the costly DTW computation. This implies that, if one insists in having the DTW as the ultimate comparison criterion, then $SDTW$ can also be used as a fast and accurate filter to discard too-dissimilar streams, thus speeding-up the overall process.

We start with a couple of preliminary definitions.

Definition 1 (Frontier) *A* frontier F *is any set of cells of the cumulative distance matrix such that for any warping path* W *it is* $W \cap F \neq \emptyset$. *The* $\llcorner$*-frontier anchored in cell* $[i,i]$ *is the set of* $2b+1$ *cells* $\llcorner[i,i] = \{[i,i],[i,i+1],\ldots,[i,i+b],[i+1,i],\ldots,[i+b,i]\}$. *The* $\urcorner$*-frontier anchored in cell* $[i,i]$ *is the set of* $2b+1$ *cells* $\urcorner[i,i] = \{[i,i],[i,i-1],\ldots,[i,i-b],[i-1,i],\ldots,[i-b,i]\}$.

Thus, any warping path (including W_{opt}) has to pass through a frontier.

Definition 2 (Boundary-relaxed DTW) *Given streams* R *and* S, *let* t_s *and* t_e *be two generic time instants. We define:*

- *The* start-relaxed DTW *(*$DTW^{\llcorner}$*) between subsequences* $R_{t_s}^{t_e}$ *and* $S_{t_s}^{t_e}$ *is the value of* $D^{\llcorner}[t_e,t_e]$, *where* $D^{\llcorner}$ *is the* start-relaxed *cumulative distance matrix initialized as follows:*

$$D^{\llcorner}[t_s,t_s+j] = d[t_s,t_s+j]; \quad D^{\llcorner}[t_s+j,t_s] = d[t_s+j,t_s] \qquad (0 \leq j \leq b) \quad (2)$$

  Thus, $DTW^{\llcorner}$ *computation proceeds as with* DTW, *yet all the cells in the* $\llcorner$*-frontier anchored in* t_s *have as value the distance between the corresponding samples (see Figure 2 (b)). This is to say that warping paths can start from* any *cell in* $\llcorner[t_s,t_s]$. *When* $t_e < t_s$ *we conventionally set* $DTW^{\llcorner}(R_{t_s}^{t_e}, S_{t_s}^{t_e}) = 0$.

[2] Note that this is the best one can achieve, since at each new time step $2b+1$ distances between samples have to be necessarily computed.

– *The* start-end-relaxed *DTW* ($DTW^{\llcorner\urcorner}$) *between subsequences* $R_{t_s}^{t_e}$ *and* $S_{t_s}^{t_e}$ *is:*

$$DTW^{\llcorner\urcorner}(R_{t_s}^{t_e}, S_{t_s}^{t_e}) = \min\{DTW^{\llcorner}(R_{t_s}^{i}, S_{t_s}^{j}) \mid [i,j] \in {}^{\urcorner}[t_e, t_e]\} \tag{3}$$

Thus, here we are also relaxing the end boundary condition. Note that for computing $DTW^{\llcorner\urcorner}(R_{t_s}^{t_e}, S_{t_s}^{t_e})$ *one still uses the same start-relaxed* $D^{\llcorner}$ *matrix used for computing* $DTW^{\llcorner}(R_{t_s}^{t_e}, S_{t_s}^{t_e})$*, and then just looks at the minimum value on the* ${}^{\urcorner}[t_e, t_e]$ *frontier (see Figure 2 (c)).*

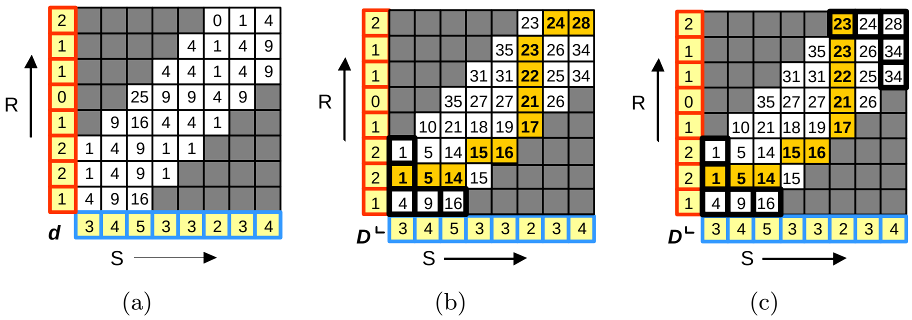


Fig. 2. (a) Distance matrix; (b) Start-relaxed *DTW*; (c) Start-end-relaxed *DTW*

The basic rationale underlying the computation of the Stream-DTW (*SDTW*) distance is to split, using frontiers, the optimal warping path of the *DTW* into 2 distinct pieces (one of them possibly null at some time instants): The 1st piece starts by spanning the whole current window of size n, and then, at each time step, progressively reduces; the 2nd piece starts empty and then progressively grows. After exactly n time steps everything starts again. For each of the 2 pieces of W_{opt} the *SDTW* provides a suitable, accurate, lower bounding measure.

We first present the formal definition of *SDTW*, a detailed explanation of how it works is provided in the Proof of Theorem 1. To stay general, we consider that we want to measure the distance between subsequences $R_{t_s}^{t_e}$ and $S_{t_s}^{t_e}$, where $t_e = kn + i$ for some positive integer k, $0 \leq i < n$, and $t_s = t_e - n + 1 = kn + i - n + 1 = (k-1)n + 1 + i$.

Definition 3 *The Stream-DTW (SDTW) distance between subsequences* $R_{t_s}^{t_e}$ *and* $S_{t_s}^{t_e}$ *is defined as:*

$$\begin{aligned} SDTW(R_{(k-1)n+1+i}^{kn+i}, S_{(k-1)n+1+i}^{kn+i}) &= DTW^{\llcorner\urcorner}(R_{(k-1)n+1}^{kn}, S_{(k-1)n+1}^{kn}) \\ &- (DTW^{\llcorner}(R_{(k-1)n+1}^{(k-1)n+1+i}, S_{(k-1)n+1}^{(k-1)n+1+i}) - d(R_{(k-1)n+1+i}, S_{(k-1)n+1+i})) \\ &+ DTW^{\llcorner}(R_{kn+1}^{kn+i}, S_{kn+1}^{kn+i}) \end{aligned} \tag{4}$$

Theorem 1 (Lower bound) *The SDTW distance is a lower bound of DTW.*

Proof. For the sake of conciseness, denote with α, β, γ and δ the 4 terms in the right-hand side of Eq. 4, so that we have to show that $\alpha - (\beta - \gamma) + \delta$ is a lower bound of DTW (see also Figure 3).

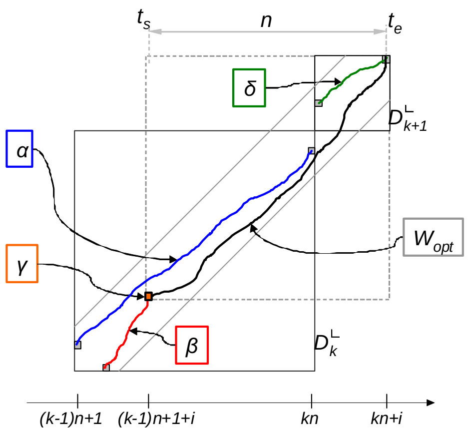


Fig. 3. How $SDTW$ works ($SDTW = \alpha - (\beta - \gamma) + \delta$)

Let W_{opt} be the optimal warping path for aligning $R^{kn+i}_{(k-1)n+1+i}$ and $S^{kn+i}_{(k-1)n+1+i}$, and consider the frontiers $\urcorner[kn, kn]$ and $\llcorner[kn+1, kn+1]$. Consider the 1st part of W_{opt}, call it $W_{opt,k}$, that ends in a cell of $\urcorner[kn, kn]$, and the 2nd part of W_{opt}, call it $W_{opt,k+1}$, that starts from a cell in $\llcorner[kn+1, kn+1]$. We claim that $\alpha-(\beta-\gamma)$ lower bounds the DTW contribution, call it DTW_k, corresponding to $W_{opt,k}$ and that δ lower bounds the component, call it DTW_{k+1}, corresponding to $W_{opt,k+1}$. Since the DTW distance between the two subsequences is $\geq DTW_k + DTW_{k+1}$ this will prove the result.
[$\alpha-(\beta-\gamma) \leq DTW_k$]. Consider the start-relaxed path, call it W_β, corresponding to β and ending in cell $[t_s, t_s]$, and the path $W_{opt,k}$, which shares cell $[t_s, t_s]$ with W_β. Counting just once the contribution of cell $[t_s, t_s]$, i.e., γ, we end up with a total cost given by $\beta - \gamma + DTW_k$ for going from $\llcorner[(k-1)n+1, (k-1)n+1]$ to $\urcorner[kn, kn]$. From the definition of $DTW^{\llcorner\urcorner}$, this cannot be less than α.
[$\delta \leq DTW_{k+1}$]. Immediate from the definition of start-relaxed DTW. $\square$

Above proof shows how we can approximate from below the DTW by splitting the optimal warping path into 2 pieces. The two frontiers we use to this purpose are by no means the only possible ones; in particular, as Figure 3 shows, a part of W_{opt} could traverse some cells, after leaving $\urcorner[kn, kn]$ and before entering $\llcorner[kn+1, kn+1]$, that $SDTW$ does not consider at all. We can prove that our arguments are still applicable should we replace $\llcorner[kn+1, kn+1]$ with the $\urcorner[kn+1, kn+1]$ frontier. For lack of space we do not enter into details here.

Turning to consider efficiency issues, now we prove that $SDTW$ is amenable to be efficiently updatable.

Theorem 2 (Complexity) *The SDTW distance can be updated in time $O(b)$ at any time step.*

Proof. Denote with α', β', γ' and δ' the new values, at time step $t_{e'} = t_e + 1 = kn + i + 1$, of the 4 terms in Eq. 4. Let $D^{\llcorner}_k$ be the start-relaxed cumulative distance matrix for time interval $[(k-1)n+1 : kn]$, and $D^{\llcorner}_{k+1}$ the one for the interval $[(kn+1 : (k+1)n]$. Let d_k and d_{k+1} be the corresponding matrices storing distances between the samples of R and S.

The steps needed to update the value of $SDTW$ include the computation of distance values for the two new samples of R and S, and the extension of matrix $D^{\llcorner}_{k+1}$ up to frontier $\urcorner[t_{e'}, t_{e'}] = \urcorner[t_e+1, t_e+1]$. Both steps require $O(b)$ time.

Consider now the case when $t_{e'} = kn + i + 1$, with $0 \le i < n-1$. We have:

- $\alpha' = \alpha$, since this term does not depend on i.
- By definition of start-relaxed DTW and of $D^{\llcorner}_k$, it is $\beta' = D^{\llcorner}_k[(k-1)n+1+i+1 : (k-1)n+1+i+1]$, i.e., computing β' costs $O(1)$. The same is clearly true for $\gamma' = d(R_{(k-1)n+1+i+1}, S_{(k-1)n+1+i+1})$.
- Finally, $\delta' = D^{\llcorner}_{k+1}[kn+i+1 : kn+i+1]$, again with cost $O(1)$.

When $i = n-1$, it is $t_{e'} = kn + (n-1) + 1 = (k+1)n$ and we have $\beta' = \gamma'$ and $\delta' = 0$ (by definition of start-relaxed DTW). The new $SDTW$ value reduces to $\alpha' = DTW^{\llcorner\urcorner}(R^{(k+1)n}_{kn+1}, S^{(k+1)n}_{kn+1})$, which, given matrix $D^{\llcorner}_{k+1}$, can be computed in $O(2b+1) = O(b)$ time. □

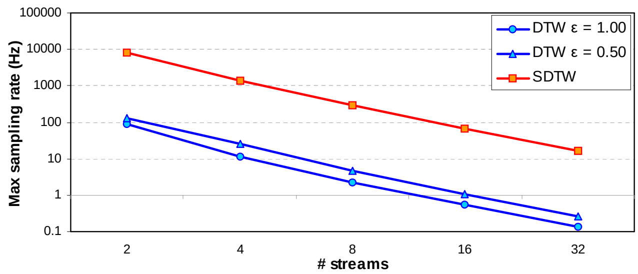
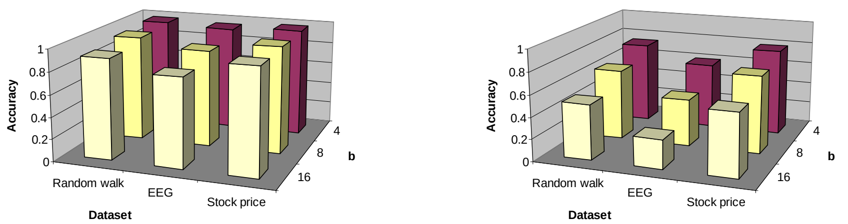


Fig. 4. Max number of streams that can be monitored with DTW and $SDTW$. Sliding window size $n = 128$. Results are averaged over the 3 datasets used in Figure 5

In order to provide a more precise characterization of the speed-up obtainable from $SDTW$, in Figure 4 we show how many streams we can monitor in real time, as compared to those manageable using DTW. More precisely, for a given number of streams, we vary their arrival frequency (sampling rate) and plot the value beyond which we cannot report results (i.e., when the distance falls below the threshold ϵ) before the next stream samples arrive. Experiments are performed on a 1.60GHz Intel Pentium 4 CPU with 512 MB of main memory and running Windows 2000 OS.

It is evident that *SDTW* outperforms *DTW* by up to two orders of magnitude. For instance, at 100 Hertz and with a sliding window of $n = 128$ samples (1.28 seconds), *SDTW* can monitor up to 16 streams (120 pairs), whereas *DTW* can only handle 2 streams.

Figure 5 (a) provides evidence of the accuracy of *SDTW* with respect to *DTW*, measured as the average of the *SDTW*/*DTW* ratio over 45 pairs of streams. For this experiment, as well as for the previous one, we use three datasets from the UCR archive [4] with very different features (shape, frequency, etc.). The accuracy of *SDTW* varies between 0.81 (EEG dataset, $b = 16$) and 0.99 (Stock price, $b = 4$). To better appreciate such values, consider that LB_Keogh, the best-so-far known method to lower bound *DTW* in the static case (see [3] for details on LB_Keogh), scores only 0.26 on EEG with $b = 16$ and 0.79 in the best case (Stock price, $b = 4$), as Figure 5 (b) demonstrates.



(a) (b)

Fig. 5. Accuracy of (a) *SDTW* and (b) LB_Keogh with respect to *DTW*

References

1. I. Bartolini, P. Ciaccia, and M. Patella. Warp: Accurate Retrieval of Shapes Using Phase and Time Warping Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(1):142–147, Jan. 2005.
2. D. J. Berndt and J. Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *AAAI 1994 Workshop on Knowledge Discovery in Databases*, pages 359–370, Seattle, WA, USA, July 1994.
3. E. J. Keogh. Exact Indexing of Dynamic Time Warping. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB'02)*, pages 406–417, Hong Kong, China, Sept. 2002.
4. E. J. Keogh and T. Folias. The UCR Time Series Data Mining Archive, 2002. Available at URL `http://www.cs.ucr.edu/~eamonn/TSDMA/index.html`.
5. C. A. Ratanamahatana and E. J. Keogh. Making Time-Series Classification More Accurate Using Learned Constraints. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, Lake Buena Vista, FL, USA, Apr. 2004.
6. H. Sakoe and S. Chiba. A Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, Feb. 1978.
7. Y. Zhu and D. Shasha. StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB'02)*, pages 358–369, Hong Kong, China, Sept. 2002.