Communication-Efficient Distributed Multiple Reference Pattern Matching for M2M Systems

Jui-Pin Wang[†] Yu-Chen Lu[†] Mi-Yen Yeh[‡] Shou-De Lin[†] Phillip B. Gibbons[§]

[†]Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
[‡]Institute of Information Science, Academia Sinica, Taiwan
[§]Intel Labs Pittsburgh, USA

Email: r01922165@ntu.edu.tw, b98902105@ntu.edu.tw, miyen@iis.sinica.edu.tw,
sdlin@csie.ntu.edu.tw, phillip.b.gibbons@intel.com

I. MsWave Algorithm and Analysis

In MsWave, we also leverage the multi-resolution property of the Haar wavelet decomposition of time series. The server P distributes the reference time series set $Q = \{S_{q1}, \ldots, S_{qn}\}$ in a level-wise manner. That is, P sends the coefficients of each $S_{qi} \in Q$ to the local machines, one level at a time starting from the highest level L. At each level, we further prune the candidates, until the final k answers are found. While similar to LeeWave at this high level, MsWave must overcome the limitations outlined in the prior section. To do this, first we derive new formulas for computing the similarity ranges of the three linkage distances between the reference set and a candidate time series (Section I-C). These ranges must be effective at pruning yet guarantee no false dismissals. Second, we devise a correct and bandwidth-efficient protocol for the data exchanges between the server and the multiple local machines (Section I-D). We present two variants: MsWave-S, which computes the bounds at the server, and MsWave-L, which computes the bounds at the local machines. Finally, we provide an analysis of the bandwidth consumption of both variants, which demonstrates the effectiveness of MsWave at reducing bandwidth (Section I-E).

A. Problem Setup

There are a query set $Q = \{q_1, q_2, ..., q_T\} \subset \mathbb{R}^D$ at the server P and a dataset $X_i \subset \mathbb{R}^D$ on each local machine M_i. For each query q_t, we want to find its k_{th} nearest neighborhood among these distributed datasets while reducing the transmission cost between P and each M_i.

B. Orthogonal Transformation

Definition 1: A matrix $W \in \mathbb{R}^{D \times D}$ is orthogonal if whose columns and rows are orthogonal vectors, i.e.

$$W^T W = W W^T = I$$

where I is the identity matrix.

Property 1: Let $x, y \in \mathbb{R}^D$, and $W \in \mathbb{R}^{D \times D}$ be an orthogonal matrix. Then,

$$Dist(x,y)^2 = \sum_{d=1}^{D} (x[d] - y[d])^2 = \sum_{d=1}^{D} (W_d x - W_d y)^2 = Dist(Wx, Wy)^2$$

where W_d is the d_{th} row of W.

C. Computation of Distance Bounds

We start from the similarity range of the distance between each individual reference time series S_{qi} to some candidate S_x. Similar to Eq. (**??**), we can derive the upper bound UB and the lower bound LB of $Dst(S_{qi}, S_x)$ as soon as all the coefficients for S_{qi} from the highest level L to the current level ℓ have been sent to the local machines, as follows:

$$LB(qi,x) = accDst^{\ell}(S_{qi}, S_x). \tag{1}$$

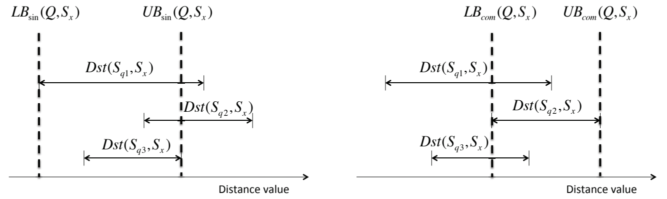
$$\begin{aligned} UB(qi,x) &= accDst^{\ell}(S_{qi}, S_x) \\ &+ \sum_{l=1}^{\ell-1} \sum_{p} ([n_{(l,p)}^{(qi)}]^2 + [n_{(l,p)}^{(x)}]^2) \times 2^l \\ &+ 2 \times \min\{\sqrt{\sum_{l=1}^{\ell-1} \sum_{p} [n_{(l,p)}^{(qi)} \times 2^l]^2 \times \sum_{l=1}^{\ell-1} \sum_{p} [n_{(l,p)}^{(x)}]^2}, \\ &\sqrt{\sum_{l=1}^{\ell-1} \sum_{p} [n_{(l,p)}^{(x)} \times 2^l]^2 \times \sum_{l=1}^{\ell-1} \sum_{p} [n_{(l,p)}^{(qi)}]^2}\}. \end{aligned} \tag{2}$$

Note that Eq. (2) is an enhanced version of the upper bound compared to that in Eq. (**??**). Because the roles of S_{qi} and S_x are interchangeable in the squared terms in Eq. (**??**), a tighter upper bound is obtained by choosing the minimum among the two choices. Our experiments will show that this subtle change noticeably improves the pruning performance. This new bound does not violate the non-increasing property proved in [1] because the smaller bound from two non-increasing bounds is chosen here.

Now we can derive the similarity range for each linkage distance defined in Definition **??**. For $d_{avg}(Q, S_x)$, the average of $Dst(S_{qi}, S_x)$ for all $S_{qi} \in Q$, we note that because the distance is non-negative, we can simply derive the new bounds as follows.

$$LB_{avg}(Q, S_x) = \frac{1}{|Q|} \sum_{i=1}^{|Q|} LB(qi, x) \tag{3}$$

$$UB_{avg}(Q, S_x) = \frac{1}{|Q|} \sum_{i=1}^{|Q|} UB(qi, x) \tag{4}$$



(a) The upper and lower bounds for $d_{sin}(Q, S_x)$.

(b) The upper and lower bounds for $d_{com}(Q, S_x)$.

Fig. 1. Upper and lower bound computations for different linkage distance measures.

For $d_{sin}(Q, S_x)$, the two bounds are:

$$LB_{sin}(Q, S_x) = \min_{1 \le i \le |Q|} LB(qi, x) \tag{5}$$

$$UB_{sin}(Q, S_x) = \min_{1 \le i \le |Q|} UB(qi, x) \tag{6}$$

Finally, for $d_{com}(Q, S_x)$, the two bounds are:

$$LB_{com}(Q, S_x) = \max_{1 \le i \le |Q|} LB(qi, x) \tag{7}$$

$$UB_{com}(Q, S_x) = \max_{1 \le i \le |Q|} UB(qi, x) \tag{8}$$

Fig. 1(a) illustrates the bounds for $d_{sin}(Q, S_x)$. As we want to choose the closest distance of a candidate time series to the reference set Q, we can set the lower (upper) bounds of $d_{sin}(Q,)$ using the smallest ones among all $LB(qi, x)$ ($UB(qi, x)$, respectively), and for $d_{com}(Q, X)$ using the largest ones among all $LB(qi, x)$ ($UB(qi, x)$), as shown in Fig. 1(b).

Now we prove the similarity ranges bounded by these lower and upper bounds will shrink as more levels of coefficients are disseminated to local machines.

Theorem 1: $UB_{avg}(Q, S_x)$ is non-increasing and $LB_{avg}(Q, S_x)$ is non-decreasing when the coefficients of $S_{qi} \in Q$ are disseminated from level ℓ to level $\ell - 1$

Proof: As argued above, it readily follows from [1] that Eq. (1) is non-decreasing and Eq. (2) is non-increasing. Therefore, $LB_{avg}(Q, S_x)$ must be non-decreasing and $UB_{avg}(Q, S_x)$ must be non-increasing as they are each just the average of a set of such non-decreasing and non-increasing bounds. ■

Theorem 2: $UB_{sin}(Q, S_x)$ is non-increasing and $LB_{sin}(Q, S_x)$ is non-decreasing when the coefficients of $S_{qi} \in Q$ are disseminated from level ℓ to level $\ell - 1$

Proof: Let $l(\ell)$ and $u(\ell)$ be the reference time series in Q that have the smallest lower bound and smallest upper bound of the similarity range to S_x at level ℓ. That is,

$$\begin{aligned} l(\ell) &= \arg\min_{1 \le i \le |Q|} LB(qi, x)|_\ell \qquad \text{and} \\ u(\ell) &= \arg\min_{1 \le i \le |Q|} UB(qi, x)|_\ell, \end{aligned}$$

where $|_\ell$ represents the corresponding bound values are derived at level ℓ. We have

$$\begin{aligned} LB_{sin}(Q, S_x)|_\ell &= LB(q_{l(\ell)}, x)|_\ell \le LB(q_{l(\ell-1)}, x)|_\ell \\ &\le LB(q_{l(\ell-1)}, x)|_{\ell-1} = LB_{sin}(Q, S_x|_{\ell-1}), \end{aligned}$$

where the first inequality holds because $l(\ell)$ is the arg min at level ℓ and the second inequality holds because Eq. (1) is non-decreasing. Thus, $LB_{sin}(Q, S_x)$ is non-decreasing.

Similarly, because Eq. (2) is non-increasing, a symmetric argument shows that $UB_{sin}(Q, S_x)$ is non-increasing. ■

Finally, the symmetry between the bounds for d_{syn} and d_{com} yields the following.

Theorem 3: $UB_{com}(Q, S_x)$ is non-increasing and $LB_{com}(Q, S_x)$ is non-decreasing when the coefficients of $S_{qi} \in Q$ are disseminated from level ℓ to level $\ell - 1$.

D. The MsWave *Protocol*

We are now ready to describe the details of how MsWave processes a distributed kNN or kFN multiple time series query in a level-wise manner and how the server P progressively prunes the candidates. We will present two schemes to solve this problem: MsWave-S and MsWave-L.

MsWave-S: Server computes the bounds. Fig. 2 presents the MsWave-S protocol. At the initial step, the server P sends the highest level-L coefficients of each S_{qi} in Q to all M local machines. Each local machine then extracts wavelet coefficients of the same level for each time series to be matched. It then returns the following numbers for each local candidate time series S_x: the level-L distance, $Dst^L(S_{qi}, S_x)$, for $i = 1$ to $|Q|$, and three other numbers that will be used by P to generate the bounds for pruning: $\sum_{l=1}^{L-1} \sum_p [n_{(l,p)}^{(x)}]^2$, $\sum_{l=1}^{L-1} \sum_p ([n_{(l,p)}^{(x)}]^2 \times 2^l)$ and $\sum_{l=1}^{L-1} \sum_p ([n_{(l,p)}^{(x)}]^2 \times 2^l)^2$. After receiving these numbers from each candidate time series, P updates the lower and upper bounds based on Eq. (1) and Eq. (2) for each candidate time series.

It then does some initial pruning to remove any candidates that cannot be among the top k neighbors. To prune candidates for kNN queries, P first sorts the candidate time series in an ascending order based on the upper bounds. Any candidate time series whose similarity lower bound is higher than the upper bound of the k^{th} time series in the sorted list cannot be in the final answer, and thus is pruned. As the bound is proved to be monotonically non-increasing from level to level, we can guarantee that there are no false dismissals under this pruning strategy. Similarly, for kFN query, any candidate time series whose similarity upper bound is smaller than the k^{th} largest lower bound cannot be in the final answer. Then, P moves to the next level.

For any given level l, P sends the level-l coefficients of each $S_{qi} \in Q$ and the ids of any pruned time series to the appropriate local machines. The local machine returns two level-specific numbers for each (remaining) candidate time series: $Dst^l(S_{qi}, S_x)$ for $i = 1$ to $|Q|$ and $\sum_p [n_{(l,p)}^{(x)}]^2$. P then uses these values to update the upper/lower bounds, always making them tighter. With the bounds of each candidate series to each reference time series, P further computes the

Procedure: MSWAVE-S for a kNN/kFN multiple time series query
Input: k, $Q = \{S_{q1}, \ldots, S_{qn}\}$, a linkage distance measure (single, average or complete)
Output: The k most similar/dissimilar time series to Q according to the designated linkage distance

The server P:	*A local machine M_i:*
1. Send coefficients of each $S_{qi} \in Q$ at level L to all M local machines.	2. For each local candidate time series, S_x, compute and return $(Dst^L(S_{qi}, S_x) \ \forall S_{qi} \in Q$, $\sum_{l=1}^{L-1} \sum_p [n_{(l,p)}^{(x)}]^2$, $\sum_{l=1}^{L-1} \sum_p ([n_{(l,p)}^{(x)}]^2 \times 2^l)$, $\sum_{l=1}^{L-1} \sum_p ([n_{(l,p)}^{(x)}]^2 \times 2^l)^2)$ to P.
3. Compute the upper and lower bounds based on Eq. (1)-(2) for each candidate time series to each reference series. Then compute the similarity range for each candidate time series to Q according to the designated linkage distance based on Eq. (3)-(8). Do the first pruning.	
4. Repeat steps 5–7 for levels $l = L-1, L-2, \ldots, 1$ until **done**{	
5. Send level coefficients of each $S_{qi} \in Q$ and the ids of any pruned candidate series to the appropriate local machines.	6. Compute and return a 2-tuple $(Dst^l(S_{qi}, S_x) \ \forall S_{qi} \in Q$, $\sum_p [n_{(l,p)}^{(x)}]^2)$ for each local candidate time series, S_x.
7. Update the upper and lower bounds based on Eq. (1)-(2) for each candidate time series to each reference series. Then update the bounds of the linkage distance based on Eq. (3)-(8). Do corresponding pruning for kNN or kFN. Set **done** to **true** if there are k candidate time series left.	
8. }	
9. Ask the appropriate machines for the contents of the final k time series.	10. If asked, send back the corresponding full time series contents.

Fig. 2. Protocol for distributed kNN/kFN query processing using MSWAVE-S.

Procedure: MSWAVE-L for a kNN/kFN multiple time series query
Input: k, $Q = \{S_{q1}, \ldots, S_{qn}\}$, a linkage distance measure (single, average or complete)
Output: The k most similar/dissimilar time series to Q according to the designated linkage distance

The server P:	*A local machine M_i:*
1. Send level L coefficients, $\sum_{l=1}^{L-1} \sum_p [n_{(l,p)}^{(qi)}]^2$, $\sum_{l=1}^{L-1} \sum_p ([n_{(l,p)}^{(qi)}]^2 \times 2^l)$, and $\sum_{l=1}^{L-1} \sum_p ([n_{(l,p)}^{(qi)}]^2 \times 2^l)^2$ of each $S_{qi} \in Q$ to all M local machines.	2. For each local candidate time series, S_x, compute the individual bounds with each reference time series using Eq. (1)-(2). Then compute and return the two linkage distances bounds based on Eq. (3)-(8).
3. Do the first pruning by sorting the upper/lower bounds of each candidate series.	
4. Repeat steps 5–7 for levels $l = L-1, L-2, \ldots, 1$ until **done**{	
5. Send level coefficients of each $S_{qi} \in Q$ and the ids of any pruned candidate series to the appropriate local machines.	6. Update the corresponding upper/lower bounds based on the computation defined in Eq. (3)-(8) and return the two linkage distance bounds for each local candidate time series.
7. Do corresponding pruning for kNN or kFN according to the updated bounds of each candidate series sent back from the local machines. Set **done** to **true** if there are k candidate time series left.	
8. }	
9. Ask the appropriate machines for the contents of the final k time series.	10. If asked, send back the corresponding full time series contents.

Fig. 3. Protocol for distributed kNN/kFN query processing using MSWAVE-L.

similarity range under the prespecified linkage distance of each series to the reference set Q based on Eq. (3)–(8). With increasingly tighter ranges, P can better prune the candidate list. The algorithm ends when there are k candidate time series remaining.

MSWAVE-L: Local machines compute the bounds. Note that with MSWAVE-S, the local machines consume bandwidth to send back the level distances of each time series to multiple reference time series, i.e., $Dst^l(S_{qi}, S_x)$ for $i = 1$ to $|Q|$, which grows linearly with Q. When $|Q|$ becomes large, the MSWAVE-S protocol might not be as efficient.

To deal with this issue, we propose another scheme, MSWAVE-L, which computes the similarity bounds under the linkage distance at the local machines. By doing so, we need not send the level distances for each reference time series, but instead only 2 single bound values for the whole query set, reducing bandwidth.

Fig. 3 presents the protocol. In the initial step, the server P sends to the local machines not only the coefficients at level L, but also three additional numbers for each reference time series $S_{qi} \in Q$, which will later enable the local machines to generate the similarity ranges: $\sum_{l=1}^{L-1} \sum_p [n_{(l,p)}^{(qi)}]^2$, $\sum_{l=1}^{L-1} \sum_p ([n_{(l,p)}^{(qi)}]^2 \times 2^l)$ and $\sum_{l=1}^{L-1} \sum_p ([n_{(l,p)}^{(qi)}]^2 \times 2^l)^2$. After receiving these values, the local machines can compute the similarity bounds of each candidate series to Q based on Eq. (3)–(8) according to different linkage distance measures. Then, each local machine sends back only the two bound values for each candidate series to the server. With the bounds of each candidate, the server P can then do corresponding pruning to tell the local machines which candidates cannot be in the final k results and can be discarded. This procedure proceeds iteratively until the final results are produced. Note that the pruning strategy is the same as that done in MSWAVE-S.

E. Analysis of Bandwidth Consumption

We will now analyze the bandwidth consumption of MSWAVE-L relative to MSWAVE-S. Suppose there are a total of s time series distributed in m local machines, and $|Q|$ reference time series. Let s_ℓ be the number of candidate time series remaining at level ℓ. We have that the difference in bandwidth between MSWAVE-S and MSWAVE-L is:

$-3|Q|m + s(4|Q|-2) + \sum_\ell 2s_\ell(|Q|-1)$, which equals

$$|Q|(4s - 3m + 2\sum_\ell s_\ell) - 2s - 2\sum_\ell s_\ell. \qquad (9)$$

The term $-3|Q|m$ refers to the three more summation values sent by P to local machines in step 1 in MsWave-L. The term $s(4|Q|-2)$ is the bandwidth saved by transmitting only the 2 bounds for each $\{Q, S_x\}$ instead of all pair-wise reference-candidate level distances and the related summation values in step 2. The final summation term describes the bandwidth saved at the following steps when more and more levels of coefficients are disseminated.

Note that Eq. (9) is always greater than zero. Because in general cases $s \geq m$ (otherwise the data do not have to be distributed to m machines), we have $|Q|(4s-3m)-2s > (|Q|-2)s \geq 0$, because $|Q| > 1$ for multiple time series, and the final term $2s(|Q|-1)$ is also greater than zero. Moreover, MsWave-L's bandwidth savings over MsWave-S increases linearly with $|Q|$.

References

[1] M.-Y. Yeh, K.-L. Wu, P. S. Yu, and M.-S. Chen, "LeeWave: level-wise distribution of wavelet coefficients for processing *k*nn queries over distributed streams," *Proc. VLDB Endow.*, vol. 1, no. 1, 2008.