APUMPEDI: Approximating Pan Matrix Profiles of Time Series Under Unnormalized Euclidean Distances by Interpolation

Jing Zhang and Daniel Nikovski
Email: {jingzhang,nikovski}@merl.com

June 29, 2022
ITISE 2022, Gran Canaria, Spain

MITSUBISHI ELECTRIC RESEARCH LABORATORIES (MERL)
Cambridge, Massachusetts, USA
http://www.merl.com

Outline

- ❑ Motivation
- ❑ Matrix Profile and Pan Matrix Profile
  - Definitions
  - Monotonicity
- ❑ APUMPEDI Algorithm
  - Description
  - Interpolation Methods
  - Numerical Results

Motivation

- The Matrix Profile (MP) is a powerful primitive for time series data mining tasks - motif discovery, discord discovery, shapelet discovery, density estimation, semantic segmentation, etc.
- Requires the practitioner to set one critical parameter - the subsequence length for similarity search
- Pan Matrix Profile (PMP) is a parameter-free version of the MP
- Expensive to compute accurate PMP
  - Exists fast algorithms to compute plain MP under normalized (for shape-based comparison) or unnormalized (for value-based comparison) distance
  - Exists fast approximate algorithms to compute PMP under normalized distance
  - Needs fast approximate algorithms to compute PMP under unnormalized distance

Matrix Profile (MP) and Pan Matrix Profile (PMP)

- Use uMP and uPMP to emphasize using *unnormalized* distance

Definitions of uMP and uPMP

- *Definition of uMP:*

The Matrix Profile (MP) of time series $X = [x_0, x_1, \ldots, x_{n-1}]$ is a new time series $Y = [y_0, y_1, \ldots, y_{n-m}]$, where

$$y_j = \min_{0 \le j' \le n-m, j' \ne j} d(X_{j,\ldots,j+m-1}, X_{j',\ldots,j'+m-1}),$$

where $d(\cdot,\cdot)$ is the ℓ_p $(1 \le p \le \infty)$ distance.

- In other words, at time index j, the value of the MP of X is the unnormalized Euclidean distance between the j th subsequence and its nearest-neighbor subsequence in X.

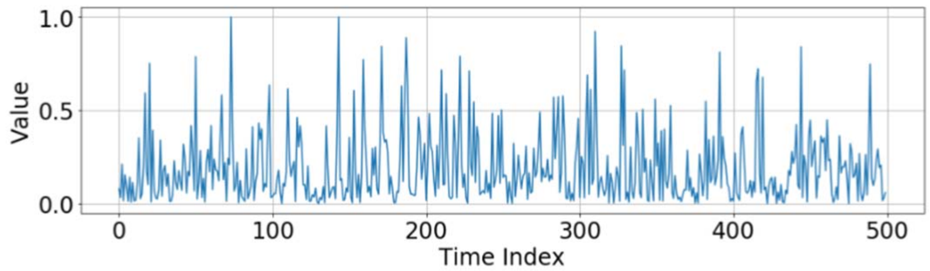
- *Definition of uPMP:*

Given a list of subsequence lengths $M = [m_0, m_1, \ldots, m_{|M|-1}]$, the Pan Matrix Profile (PMP) of time series $X = [x_0, x_1, \ldots, x_{n-1}]$ is an $|M| \times n$ matrix P with each row filled by a Matrix Profile of X, which corresponds to a specific subsequence length; in particular,

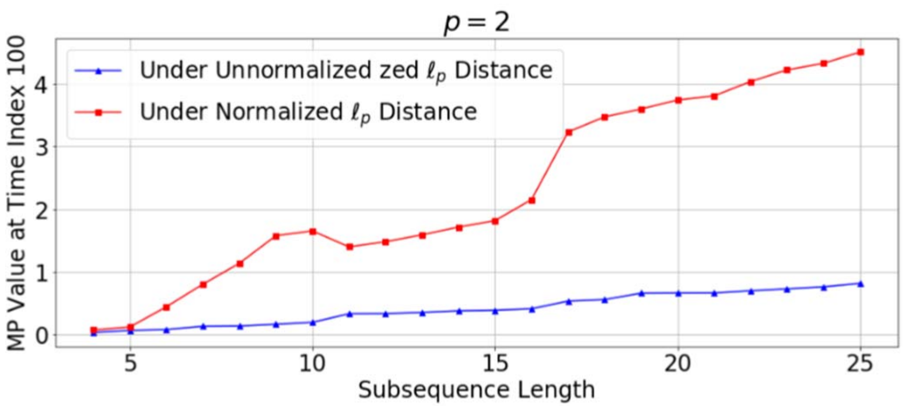
$$P[i,j] = \min_{0 \le j' \le n-m_i, j' \ne j} d(X_{j,\ldots,j+m_i-1}, X_{j',\ldots,j'+m_i-1}),$$

$\forall 0 \le i \le |M|-1, 0 \le j \le n-m_i$, where m_i is the subsequence length corresponding to the i-th row.

Monotonicity of the uPMP with respect to the subsequence lengths



(a)



(b)

Fig. 1: (a) A synthesized time series; (b) MP values at time index 100 vs. subsequence lengths of the synthesized time series.

- In Fig. 1 we plot its MP values under both the normalized and unnormalized l_2 distances at the 100th time instance, versus the subsequence lengths.
- It is seen that, under normalized l_2 distance, the MP values are not consistently increasing with respect to the subsequence lengths (there is an obvious decrease when the subsequence length increases from 10 to 11).

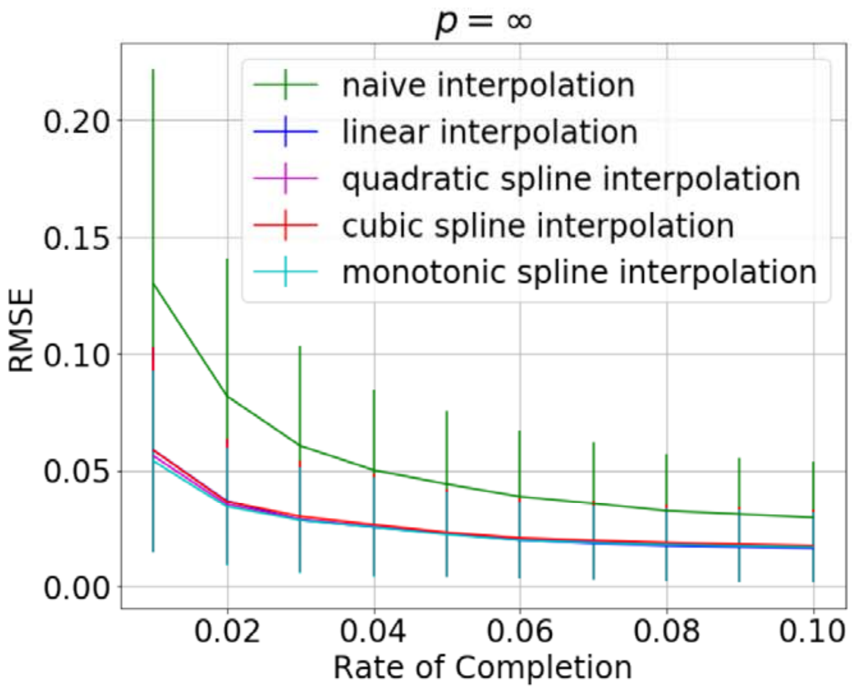
APUMPEDI Algorithm

Algorithm 1 Approximating Pan Matrix Profile under Unnormalized Euclidean Distance by Interpolation

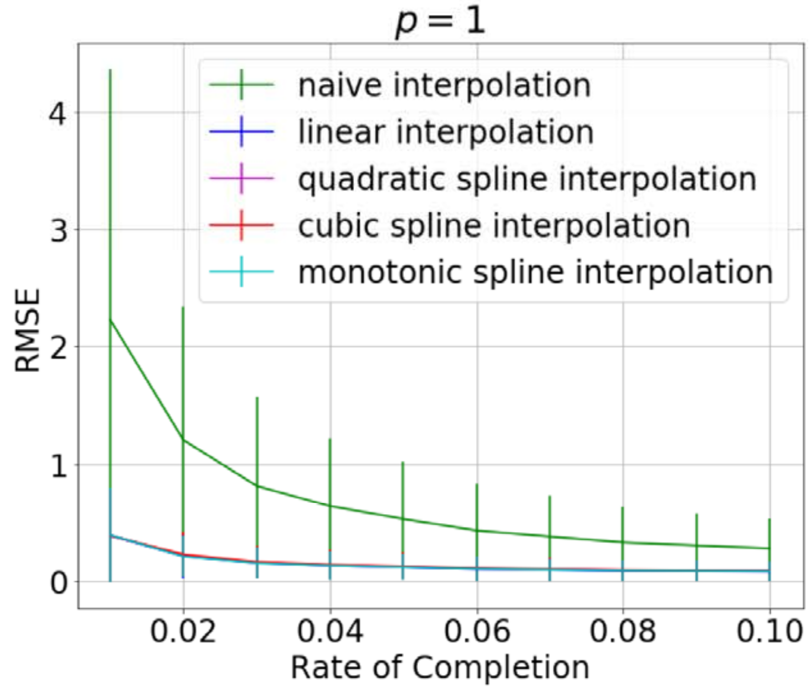
procedure APUMPEDI(X, L, U, S, θ, p)
  $n \leftarrow$ length of X
  $M \leftarrow \text{range}\big(L, U+1, S\big)$
  $P \leftarrow |M| \times n$ matrix of NaN's
  $M' \leftarrow \text{range}\big(L, U+1, \lfloor \frac{1}{\theta} \rfloor \cdot S\big)$
  for m in M' **do**
    $i \leftarrow m@M$
    $P[i,:] \leftarrow \texttt{uMP}(X, m, p)$
  end for
  $x_{vec} \leftarrow M'$
  for j in range$(0, n, 1)$ **do**
    $y_{vec} \leftarrow [P[m@M, j]$ for m in $M']$
    $f(\cdot) \leftarrow \texttt{Interpolate}(x_{vec}, y_{vec})$
    for m in $M \setminus M'$ **do**
      $i \leftarrow m@M$
      $P[i,j] \leftarrow f(m)$
    end for
  end for
  return P
end procedure

- Algorithm description
  - L(resp., U): minimum (resp., maximum) subsequence length
  - S: step size
  - $\theta \in (0,1]$: completion rate for the PMP computation
  - apply a dynamic programming based algorithm for the uMP subroutine (Line 8), which computes the MP of a given time series X, for a given subsequence length m, under the unnormalized l_p distance.
  - Lines 10 through 18 describe the interpolation procedure
- Interpolation Methods
  - Naive Interpolation: simply returns the previous value of the point
  - Linear Interpolation: concatenation of linear interpolants between each pair of data points
  - Spline Interpolation: fits low-degree polynomials (called splines) to small subsets of the data points
  - Monotonic Cubic Interpolation: apply the Piecewise Cubic Hermite Interpolating Polynomial

Numerical Results (I)



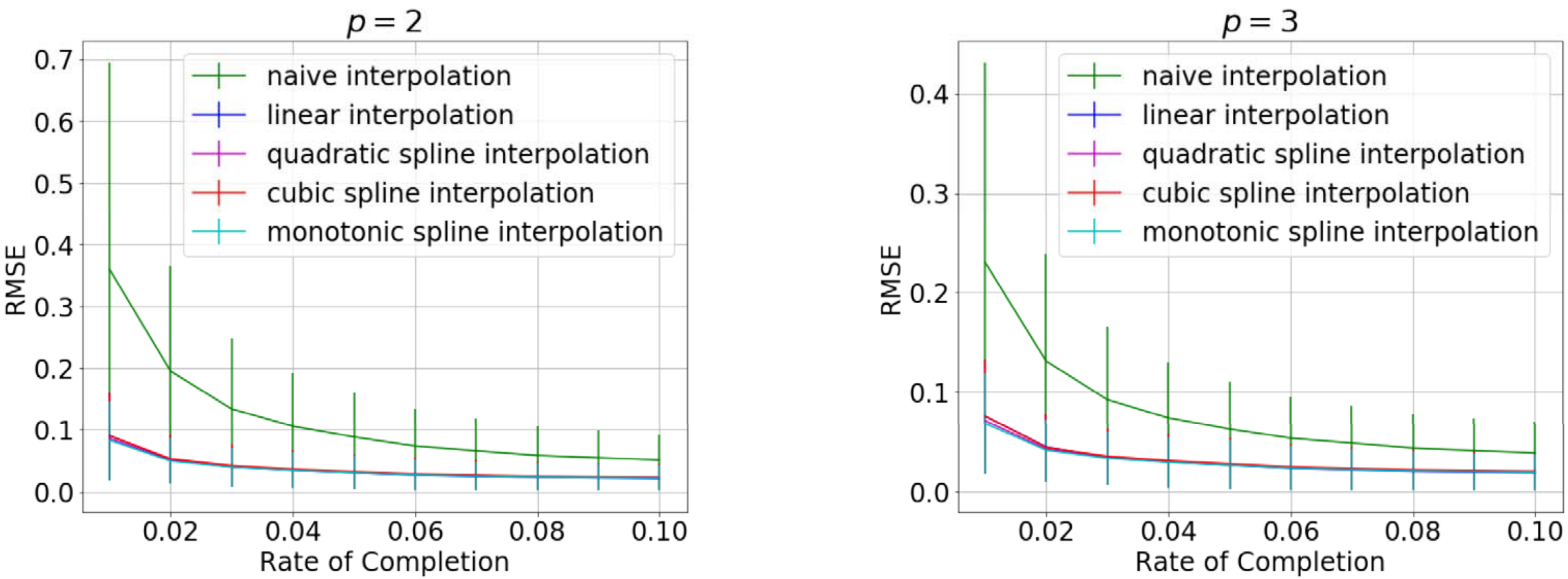
(a)



(b)

Fig. 3: A comparison of the RMSE versus Rate of Completion (i.e., θ) curves of the APUMPEDI algorithm with different interpolation methods for PMP approximation under unnormalized ℓ_p distance, where $p = \infty, 1, 2, 3,$ respectively. Here we use the time series of the Mars Curiosity Rover $A1 - A3, D1 - D3, E1 - E3, T1 - T3,$ $P1 - P3$ The RMSE scores are obtained by averaging over the results from the 15 real-world time series for each rate of approximation, and we also show the standard deviations for these scores as error bars.

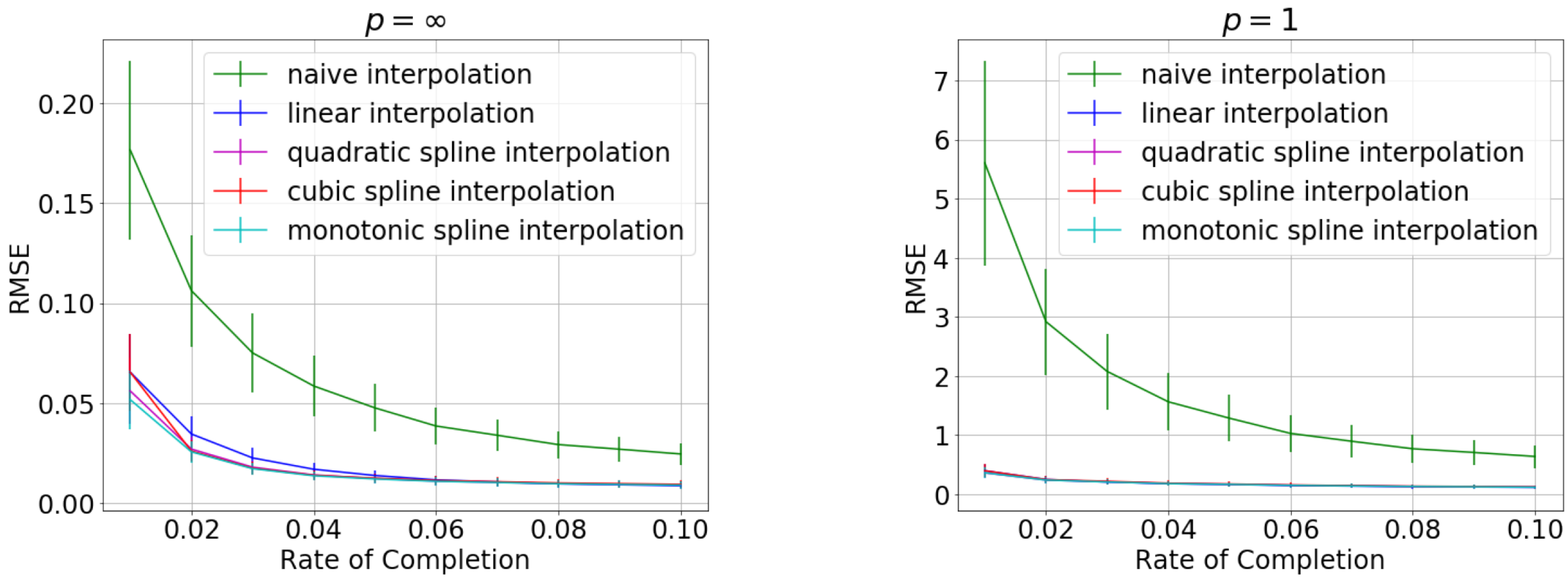
Numerical Results (I)(cont.)



(c) (d)

Fig. 3: A comparison of the RMSE versus Rate of Completion (i.e., θ) curves of the APUMPEDI algorithm with different interpolation methods for PMP approximation under unnormalized ℓ_p distance, where $p = \infty, 1, 2, 3,$ respectively. Here we use the time series of the Mars Curiosity Rover $A1 - A3, D1 - D3, E1 - E3, T1 - T3,$ $P1 - P3$ The RMSE scores are obtained by averaging over the results from the 15 real-world time series for each rate of approximation, and we also show the standard deviations for these scores as error bars.

Numerical Results (II)

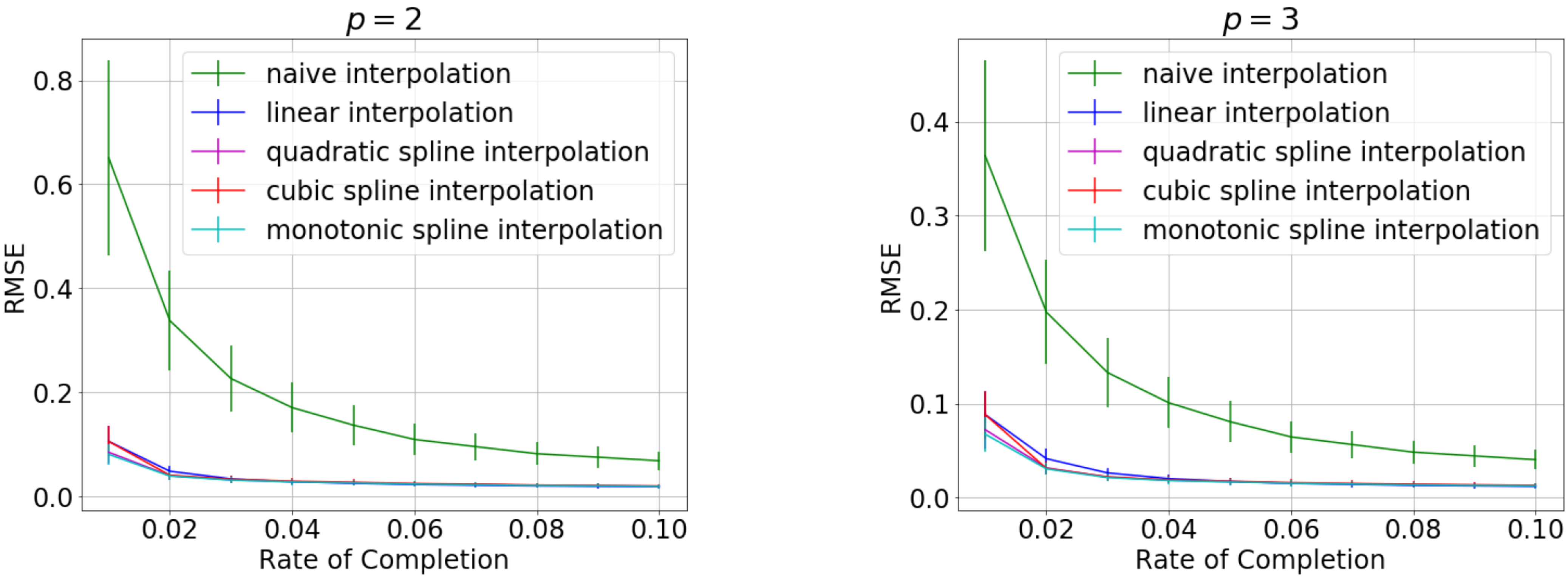


(a) (b)

Fig. 4: A comparison of the RMSE versus Rate of Completion (i.e., θ) curves of the APUMPEDI algorithm with different interpolation methods for PMP approximation under unnormalized ℓ_p distance, where $p = \infty, 1, 2, 3,$ respectively. The RMSE scores are obtained by averaging over the results from the 15 synthesized time series for each rate of approximation, and we also show the standard deviations for these scores as error bars.

MITSUBISHI ELECTRIC
Changes for the Better

Numerical Results (II)(cont.)



(c) (d)

Fig. 4: A comparison of the RMSE versus Rate of Completion (i.e., θ) curves of the APUMPEDI algorithm with different interpolation methods for PMP approximation under unnormalized ℓ_p distance, where $p = \infty, 1, 2, 3,$ respectively. The RMSE scores are obtained by averaging over the results from the 15 synthesized time series for each rate of approximation, and we also show the standard deviations for these scores as error bars.

Observations

- Relatively speaking, in all scenarios, the naive interpolation method would lead to the worst RMSE scores, while the other interpolation methods would lead to close RMSE scores, especially for the case where $p = 1$ (i.e., under the l_1 distance).
- Moreover, the smaller the rate of completion, the more the advantages (resp., disadvantages) of the monotonic spline interpolation method (resp., the linear interpolation method) over other alternatives.
- Based on the above observations, we suggest using the PCHIP monotonic spline interpolation in practice, especially when needing to set an extremely small completion rate.

Observations (cont.)

- At any time, the algorithm could be terminated, resulting in an approximation of the exact PMP.

- For a time series with length 4000, if setting $L = 3, U = 500, S = 1, \theta$= 0.05, the APUMPEDI algorithm implemented in Python can finish running within 10 minutes on a desktop computer with an Intel(R) Core(TM) i7-4770 CPU and 16 GB of system memory.

- Our experimental results suggest that, in practice, we could set θ to relatively small values (say 0.01, 0.05), while still achieving good enough performance (in terms of RMSE).

MITSUBISHI ELECTRIC

Changes for the Better