Temporal Streaming Batch Principal Component Analysis for Time Series Classification (Student Abstract)

Enshuo Yan, Huachuan Wang, Weihao Xia

Qingdao Innovation and Development Center, Harbin Engineering University, Qingdao, China
{yanenshuo, hcwang, weihao.xia}@hrbeu.edu.cn

Abstract

In multivariate time series classification, although current sequence analysis models have excellent classification capabilities, they show significant shortcomings when dealing with long sequence multivariate data. This paper focuses on optimizing model performance for long-sequence multivariate data by mitigating the impact of extended time series and multiple variables on the model. We propose a principal component analysis (PCA)-based temporal streaming compression and dimensionality reduction algorithm for time series data (temporal streaming batch PCA, TSBPCA), which continuously updates the compact representation of the entire sequence through streaming PCA time estimation with time block updates, enhancing the data representation capability of a range of sequence analysis models.We evaluated this method using various models on five datasets, and the experimental results show that our method demonstrates outstanding performance in both classification accuracy and time efficiency.

Introduction

As practical demand continues to grow, the scope of time series data has expanded rapidly in terms of quantity and the increasing complexity of inherent characteristics, such as its multivariate structure and extended time spans. it demands more storage, computational capacity, and advanced techniques to capture variable interactions. While models like RNNs and Transformers have enhanced sequence modeling, they still continue to face challenges with long sequences, including vanishing gradients and inefficiencies in attention mechanisms (Nokleby, Raja, and Bajwa 2020).

Traditional PCA methods, based on eigen-decomposition or SVD, are unsupervised techniques that compute principal components from the entire dataset's covariance matrix. However, these methods are inefficient for large-scale or streaming data. To address this limitation, in 1982, Oja introduced streaming PCA (Oja and Karhunen 1985), which uses a batch stochastic power method for PCA. Yang et al. (Yang, Hsieh, and Wang 2018) later improved this with hist-PCA by incorporating historical data for greater stability. However, hist-PCA still struggles to handle time series data due to its lack of temporal awareness.

Algorithm 1: Temporal Streaming Batch PCA

Input: Sequence of time series data $\{X_1, \ldots, X_N\}$, data size B, time batch T, starting vector $H \sim N(0, I_{d\times d})$, $H = Q_1 R$ (QR-decomposition)

Output: Compact representation of data $\widehat{X} = \{X_1, \ldots, X_t\}$

while Q_1 not converge **do**
  $W_1 \leftarrow Q_1 + \frac{1}{B} X_1^T X_1 Q_1$
  $W_1 = Q_1 R_1$
end while
$\lambda_\tau = \|W_1[:\tau]\|_2$ for $\tau = 1, \ldots, k$
$\Lambda_1 = \text{diag}(\lambda_1, \ldots, \lambda_\tau)$
for $i = 2$ to N/T **do**
  $X_i = X[i - 1 \times T : i \times T]$
  for $j = 1$ to t **do**
    while Q_j not converge **do**
      $W_j \leftarrow \frac{j-1}{j} Q_{j-1} \Lambda_{j-1} Q_{j-1}^T Q_j + \frac{1}{j} \cdot \frac{1}{B} X_j^T X_j Q_j$
      $W_j = Q_j R_j$
    end while
    $\lambda_\tau = \|W_j[:\tau]\|_2$ for $\tau = 1, \ldots, k$
    $\Lambda_j = \text{diag}(\lambda_1, \ldots, \lambda_\tau)$
  end for
  $\widehat{X}_i = X_i Q_j$
end for
return $\widehat{X}$

We extend this method to incorporate the temporal dimension, adapting it to the characteristics of time series data. By incrementally updating along the data's temporal flow, our approach retains temporal dependencies while extracting principal directions among multiple variables, resulting in a unique data representation.

Our Method

We consider implementing stepwise dimensionality reduction of data through time-dependent estimation based on streaming PCA, focusing on the advantage of streaming PCA in processing data sequentially. By leveraging incremental iteration, each new data point is linked to previous ones. Specifically for time series data, our approach sequentially feeds data into PCA, rather than randomly sampling independent instances. This fully captures and retains the

Dataset	Method	LSTM		Transformer		Informer		iTransformer		TimesNet	
		Acc	Time	Acc	Time	Acc	Time	Acc	Time	Acc	Time
UWave	TSBPCA	**57.9**	**9.08**	**87.2**	13.12	85.6	**13.94**	76.9	**12.10**	82.2	20.97
	w/o PCA	50.9	16.72	86.6	**12.94**	**86.2**	14.40	76.9	12.48	**85.6**	**19.76**
HB	TSBPCA	**74.8**	**16.26**	76.6	**16.20**	**78.0**	**17.22**	75.1	**15.35**	75.1	17.84
	w/o PCA	71.9	24.80	**77.6**	16.80	77.6	18.13	**76.6**	15.96	**76.1**	**17.49**
SR1	TSBPCA	**87.5**	**35.96**	90.1	**18.27**	87.7	**20.14**	**85.3**	**17.82**	85.3	48.60
	w/o PCA	86.8	96.58	**90.4**	33.73	**90.1**	21.59	90.4	18.51	**90.1**	**43.69**
SR2	TSBPCA	**54.8**	**32.29**	**57.2**	**15.50**	**57.2**	19.09	**59.4**	**14.68**	**53.9**	**42.79**
	w/o PCA	53.9	85.28	51.7	39.02	54.4	**18.75**	54.4	16.68	51.7	55.34
EC	TSBPCA	**33.1**	**57.45**	**32.3**	**19.88**	**30.4**	**20.20**	**30.8**	19.88	**31.2**	**41.54**
	w/o PCA	31.9	175.68	27.4	94.92	29.7	25.56	27.0	**18.81**	30.0	52.86
AVERAGE	TSBPCA	**61.6**	**30.21**	**68.7**	**16.59**	**67.8**	**18.12**	**65.5**	**15.97**	65.5	**35.75**
	w/o PCA	59.1	79.81	66.7	39.48	67.6	19.69	65.1	16.49	**66.7**	37.83

Table 1: Performance comparison of different methods on various datasets. B=16. The TSBPCA parameters T and K are selected based on optimal performance.

temporal dependencies, more accurately reflecting the dynamic characteristics of time series.

At time step 1, we receive the first time batch X_1. Since no past temporal information is available, we use a randomly initialized $d \times k$ matrix Q_1 to compute the matrix of k feature vectors for the first time point.

At time step 2, we process data X_j for the next time point. Incorporating past temporal information, we update W_1 and use the estimator $\frac{1}{2}\left(\frac{1}{B}X_2^T X_2 + W_1 W_1^T\right)$, which considers historical data. For time step j $(1 < j < t)$, the latest covariance matrix estimate is obtained using data from the previous $j-1$ steps. We assign equal weights to the data within each batch, updating W_j as follows:

$$W_j \leftarrow \frac{j-1}{j} Q_{j-1} \Lambda_{j-1} Q_{j-1}^T Q_j + \frac{1}{j}\frac{1}{B} X_j^T X_j Q_j \quad (1)$$

At time t, the batch output value Q_i is used as the time projection matrix for the batch data, completing the compact representation for this time point. Q_i then replaces the initial random values as the historical information for the next batch, continuing the iterative process until all time points are processed, resulting in the desired compact representation.

Experiments And Results

We selected five real-world datasets with varying sequence lengths: UWaveGestureLibrary (UWave), Heartbeat (HB), SelfRegulationSCP1 (SCP1), SelfRegulationSCP2 (SCP2), and EthanolConcentration (EC), containing 3 to 61 variables and sequence lengths from 300 to 1800. These datasets enabled a comprehensive evaluation of sequence lengths. Our method was tested across five models: LSTM, Transformer, Informer, iTransformer, and TimesNet.

The experimental results, detailed in Table 1, compare outcomes using the TSBPCA algorithm against those without enhancement methods.

The results show that the TSBPCA + LSTM model achieves significant improvements across all five datasets, with an average accuracy increase of 4.23% and a 62.15% boost in time efficiency. This enhancement is attributed to the RNN's inherent adaptability to sequential data due to its cyclic structure, allowing it to effectively adapt to our method. For other TSBPCA-enhanced models, the improvements are less consistent, varying across datasets, but they still generally outperform traditional methods in both accuracy and time. Notably,The method shows significant improvements on the SR2 and EC datasets, with accuracy increasing by up to 18% and time efficiency improving by up to 80%, highlighting its suitability for long sequence data.

Conclusions

In summary, we introduce the TSBPCA method that generates low-dimensional approximations of data through time estimation in streaming PCA, significantly optimizing the performance of multivariate time series classification models when handling long sequences. The experimental results demonstrate that the method improves classification accuracy and efficiency across multiple real-world datasets and models, particularly showing stronger adaptability to long sequence data. Our dual-iteration internal data reconstruction mechanism may still present limitations in processing highly complex time series or datasets with extreme variability. Moving forward, we aim to refine the reconstruction process and exploring more efficient algorithm implementations.

References

Nokleby, M.; Raja, H.; and Bajwa, W. U. 2020. Scaling-up distributed processing of data streams for machine learning. *Proceedings of the IEEE*, 108(11): 1984–2012.

Oja, E.; and Karhunen, J. 1985. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of mathematical analysis and applications*, 106(1): 69–84.

Yang, P.; Hsieh, C.-J.; and Wang, J.-L. 2018. History PCA: A new algorithm for streaming PCA. *arXiv preprint arXiv:1802.05447*.