Footbar

A Two-Fold Approach to Handling Time-Series Data: GANs and Random Forest

Author: :

Amine ELKARI
amine.elkari@polytechnique.edu

October 1, 2023

1 Introduction:

Dealing with time-series data presents unique challenges, especially when the goal is to recreate complex patterns and classify labels accurately. To tackle this problem effectively, I adopted a two-fold approach. My strategy involved regenerating time series data and assigning labels to their corresponding norms. In this article, I'll delve into the methods I employed to achieve this, namely Time-series Generative Adversarial Networks (GANs) and Random Forest, while also touching on a less effective approach I experimented with – Dynamic Time Warping with k-Nearest Neighbors (kNN).

2 Time-series Generative Adversarial Networks (GANs):

This is the important part : I tried ti implement the GANs Time series paper approach :
"**Time-series Generative Adversarial Networks**"

For further references see The paper

To reconstruct and regenerate time series data, I turned to Time-series Generative Adversarial Networks (GANs). GANs are a powerful class of machine learning models specifically designed for generating data samples that resemble a given distribution. In my case, GANs were instrumental in recreating the intricate patterns and structures present in football game time series data.

The GAN framework consists of two components: a generator and a discriminator. The generator attempts to create synthetic time series data that is indistinguishable from the real data, while the discriminator tries to differentiate between real and synthetic samples. Through an adversarial training process, the generator becomes adept at generating realistic time series data, which is crucial for subsequent analysis and classification.

3 Random Forest for Label Classification:

After successfully regenerating the time series data, the next step was to classify the labels associated with these time series. For this task, I opted for a Random Forest classifier.

3.1 Data Preparation:

First, I need to prepare my data. I have already generated synthetic time series data using Time-series Generative Adversarial Networks (GANs). Each data point in this synthetic dataset should have associated labels that describe the game state (norm).

3.2 Random Forest Training:

- Features and Labels: In order to train my Random Forest model, I'll treat the real time series data as my features and the associated labels as my target variable. The features are essentially the patterns and characteristics I've captured in the regenerated time series data.

- Training Set: I'll split my dataset into a training set and a separate testing.
- Model Training: Now, I feed my training data (the real time series and their corresponding labels) into the Random Forest algorithm. The Random Forest algorithm will create multiple decision trees, each trained on different subsets of my data. These decision trees will collectively make decisions about how to classify the labels based on the patterns in the synthetic data.

3.3 Model Evaluation:

After training my Random Forest model, it's essential to evaluate its performance on the testing set. I can use metrics such as imbalanced accuracy (I have an imbalanced data classification), and also metrics report precision, recall, F1-score, to determine how well my model is classifying the labels.

3.4 Recreating the Game:

Once my Random Forest model is trained and performing well, I can use it to classify labels for regenerated time series data. These labels can represent different aspects of the football game.

- Label Assignment: For each generated time series, I input it into my trained Random Forest model. The model will predict the label(s) associated with that time series based on the patterns it has learned during training.
- Game Reconstruction: With the labels assigned to each time series, I can now use this information to reconstruct the football game. The labels serve as instructions or descriptions of what's happening in the game at different time points. By piecing together these labeled time series, I can recreate the flow and events of the football game based on the output of my Random Forest model.

4 Dynamic Time Warping with k-Nearest Neighbors (kNN):

In my pursuit of finding the most suitable approach for label classification, I experimented with Dynamic Time Warping (DTW) in conjunction with k-Nearest Neighbors (kNN). DTW is a technique used to measure the similarity between two time series by accounting for variations in their temporal alignment.

While DTW with kNN is a valuable method for time series classification, it yielded results that were less satisfactory compared to Random Forest. This outcome underscores the importance of selecting the right tools and techniques for specific tasks. In this instance, the complexity of football game time series data may have posed challenges that made Random Forest a more suitable choice.

5 Conclusion:

The two-fold approach of utilizing Time-series Generative Adversarial Networks (GANs) for time series regeneration and Random Forest for label classification proved to be a robust solution for handling complex football game time series data. The combination of data generation and classification techniques ensured accurate analysis and decision-making. While Dynamic Time Warping with k-Nearest Neighbors was an interesting alternative, its inferior performance in this context highlighted the significance of choosing the right methodology for the task at hand.

For the git repository : Repo_of_the_code