Structure-Preserving Oversampling for Imbalanced Multivariate Time Series

Abstract

Creating synthetic minority samples for highly imbalanced multivariate time series is a challenging task due to the fact that adjacent data points along temporal and feature dimensions can be highly correlated. Traditional interpolation-based oversampling techniques such as Synthetic Minority Oversampling does not preserve the underlying covariance structure of the time series data and thus can compromise classification performance. We propose a novel framework addressing the task of oversampling highly imbalanced multivariate time series data with the goal of preserving the underlying covariance structure. The learning framework is based on modeling the minority class distribution as a Gaussian Mixture Model. An optimization algorithm, called Structure-Preserving Expectation-Maximization Oversampling (SEMO), is proposed. Given initial samples created via a structure-preserving oversampling method, SEMO jointly optimizes the minority class distribution based on Expectation-Maximization and updates the synthetic minority data with those from the optimized distribution. Based on extensive numerical experiments, the proposed framework demonstrates improvement in classification performance across multiple multivariate time series benchmark datasets.

1 Introduction

Time series classification is an important statistical learning task with a wide range of applications in fields such as science, finance, and healthcare. Time series data is defined as an ordered set of real-valued variables that are sampled from a continuous signal, which can be in the time or spatial domain [Wei and Keogh, 2006]. Multivariate time series refers to the time series data where there is more than one feature at any single point of time, for example, in human physiological vitals, heart and respiratory rates are two different features that vary over time simultaneously [Runger, 2015]. For univariate time series, due to the sequential nature, adjacent data points can be highly correlated with each other. For multivariate time series data, not only can data points correlate along the temporal dimension, they may also correlate in the feature dimension. For example, a person's respiratory rate should correlate with the heart rate at any given time point.

In binary classification, imbalanced data refers to the fact that there are far fewer minority class samples than majority class samples. Imbalanced samples naturally pose challenges to a classification model since it is difficult for the classifier to properly learn the characteristics of the minority samples. Existing approaches that address an imbalanced learning task generally fall into two categories, data-level [Chawla *et al.*, 2002; He *et al.*, 2008; Estabrooks *et al.*, 2004; Batista *et al.*, 2004; Han *et al.*, 2005; Liu *et al.*, 2009] or algorithm-level [He and Garcia, 2009; Sun *et al.*, 2007], alternatively, a combination of both can be used [Chawla *et al.*, 2003; Guo and Viktor, 2004; Chen *et al.*, 2010]. A data-level approach aims to re-establish the class balance by oversampling the minority samples, or downsampling the majority samples. Typically, oversampling is preferred because downsampling suffers the risk of losing important information presented in the data. An algorithm-level approach usually addresses the issue by manipulating the loss function and incorporating learning parameters, such as class-dependent weights. Our work falls into the category of a data-level approach as we aim to optimally establish the class balance for imbalanced multivariate time series by oversampling.

Given the special characteristics of multivariate time series, generating synthetic minority data that can capture the underlying correlations along the temporal dimension, feature dimension and cross feature-temporal dimension is not a trivial task. Existing data-level approaches generally take on two directions, interpolation-based techniques and structure-preserving techniques. The interpolation-based techniques are represented by Synthetic Minority Oversampling (SMOTE) [Chawla *et al.*, 2002] and Adaptive Synthetic Sampling (ADASYN) [He *et al.*, 2008] which create synthetic minority samples by interpolating between a selected minority sample and its nearest neighbors. Interpolation-based methods are applicable to any type of data, not only time series. When applied to time series, the synthetic samples created do not preserve the underlying correlations along the temporal dimension because the oversampling algorithm treats each time point as an independent feature. On the other hand, structure-preserving oversampling is proposed specifically for the task of univariate time series oversampling with the goal of preserving the correlation along the temporal dimension [Cao *et al.*, 2013; Cao *et al.*, 2014]. However to the best of our knowledge, there is no existing framework that directly addresses minority oversampling of multivariate time series based on a structure-preserving approach. The goal of this study is to propose a learning framework that directly addresses the minority oversampling task for multivariate time series data via an approach that combines the structure-preserving oversampling and a joint optimization algorithm.

The proposed framework is based on modeling the minority class distribution of multivariate time series as a Gaussian Mixture Model. For each Gaussian model in the mixture, we first create initial synthetic minority samples by drawing from a multivariate Gaussian distribution based on the regularized covariance matrix. With these initial synthetic samples, the proposed **S**tructure-preserving **E**xpectation-**M**aximization **O**versampling (SEMO) algorithm jointly optimizes the minority class distribution and replaces a portion of the synthetic samples with those from the optimized distribution. Our major contributions are as follows

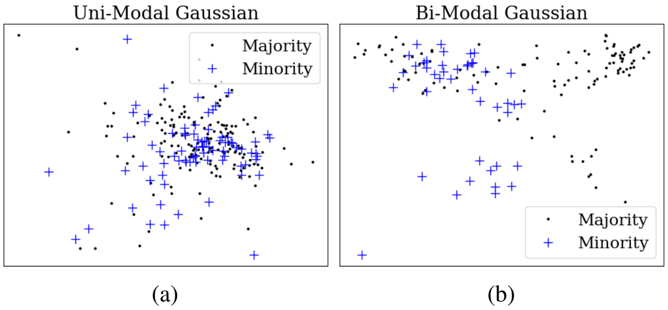
- We propose, to the best of our knowledge, the first learning framework to directly address the task of oversampling imbalanced multivariate time series data. Previous work [Cao *et al.*, 2013; Cao *et al.*, 2014; Cao *et al.*, 2011] exclusively focused on univariate time series. Our work is not a simple extension but with different model and algorithm that specifically target the multivariate time series.
- We propose to model the minority class of multivariate time series with a Gaussian Mixture Model and use structure-preserving oversampling to create synthetic minority samples.
- We propose the SEMO algorithm based on the EM framework that combines optimization of minority class samples and oversampling the minority class from the optimized distribution. The proposed learning framework demonstrated a better classification performance compared to existing oversampling techniques, across multiple multivariate time series benchmark datasets.

2 Structure-Preserving Oversampling Framework

The proposed learning framework is based on modeling the minority samples as a Gaussian Mixture Model. The ultimate goal of the learning framework is to generate synthetic minority samples from an optimized distribution based jointly on original minority data and the previously generated synthetic minority samples with the structure-preserving oversampling method. The overall flow is as follows: upon initial inspection of the minority class we determine the number of Gaussian models, then for each Gaussian model, we create initial synthetic samples using the enhanced structure-preserving oversampling method (detailed in Section 2.2). Then, given the original minority samples and the initial synthetic samples generated, we run the proposed SEMO algorithm that jointly optimizes the sample distribution of both the original minority data and the initial generated synthetic samples. The optimization is is performed concurrently with updating of a portion of the synthetic minority data with the samples drawn from the latest fitted distribution. We stop the process after a certain number of iterations based on classification performance on the validation data set.

2.1 Model

Previous works are based on the assumption that minority class samples always follow a single Multivariate Gaussian model [Cao *et al.*, 2013; Cao *et al.*, 2014]. Our observations with multivariate time series data suggest that for many datasets, the minority class samples do not always follow a single Multivariate Gaussian Model, instead the minority class samples often manifest themselves as a mixture of Gaussian Models. Figure 1 shows majority and minority class distribution for two datasets from the University of California Riverside (UCR) multivariate time series repo [Bagnall *et al.*, 2018]. The visualizations show the projections of samples onto the first two eigen vectors (X and Y values). The minority samples are represented by blue crosses and the majority samples are represented by black dots. The visualizations illustrate cases where minority samples manifest themselves as a uni-modal Gaussian distribution (Part (a)), and bi-modal Gaussian mixtures (Part (b)). The minority sample distribution following a multi-modal distribution is understandable because in reality, the minority class often represents some rare event such as the presence of engine failure and it is possible that there are multiple failure mechanisms, manifesting themselves as different Gaussian models. For this reason, we decided to model the minority class distribution as a Gaussian Mixture Model.



(a)

(b)

Figure 1: Examples of minority class distribution. Part (a) shows dataset FingerMovements where the minority class manifests a uni-modal Gaussian. Parts (b) shows the datasets RacketSports, where the minority class manifests bi-modal Gaussian distribution. The X and Y-axes correspond to the projection onto the first two eigenvectors in the eigen signal space.

2.2 Structure-Preserving Oversampling

Multivariate Time Series (MTS) sample can be expressed as $\boldsymbol{X} = \{x_{ij}\}, 1 \leq i \leq K, 1 \leq j \leq T$, where K is the number of features, T is the length of temporal dimension. One sample $\boldsymbol{X}$ can be represented by a $K \times T$ matrix. In a binary classification setting, the dataset is constituted of the minority class and the majority class, which are represented by 3-way tensors of size $P \times K \times T$ and $N \times K \times T$, respectively. Here P refers to the number of minority class samples and N refers to the number of majority class samples. For the imbalanced classification problem we are addressing, we ensure $P \ll N$.

The algorithm first flattens the minority class data and majority class data by converting the 3-way tensor into matrices of shape $P \times KT$ and $N \times KT$, each sample is now represented by a vector $\boldsymbol{x_{pi}} \in R^{KT}$, where $i = 1, \cdots, P$ and $\boldsymbol{x_{nj}} \in R^{KT}$, where $j = 1, \cdots, N$, the first subscript p or n is to note whether this sample belongs to the minority class or majority class. Given the matrix representation of the data, we first compute the total covariance matrix

$$\boldsymbol{W_{total}} = \frac{1}{P+N}[\sum_{i=1}^{P}(\boldsymbol{x_{pi}} - \overline{\boldsymbol{x}})(\boldsymbol{x_{pi}} - \overline{\boldsymbol{x}})^T + \sum_{j=1}^{N}(\boldsymbol{x_{nj}} - \overline{\boldsymbol{x}})(\boldsymbol{x_{nj}} - \overline{\boldsymbol{x}})^T] \quad (1)$$

where $\overline{\boldsymbol{x}}$ is the mean vector

$$\overline{\boldsymbol{x}} = \frac{1}{P+N}[\sum_{i=1}^{P}\boldsymbol{x_{pi}} + \sum_{j=1}^{N}\boldsymbol{x_{nj}}] \quad (2)$$

Then we perform eigen decomposition of the total covariance matrix $\boldsymbol{W_{total}}$

$$\boldsymbol{A} = \boldsymbol{L}^T \cdot \boldsymbol{W_{total}} \cdot \boldsymbol{L} \quad (3)$$

where $\boldsymbol{L} = [\boldsymbol{l_1}, \cdots, \boldsymbol{l_j}, \cdots \boldsymbol{l_{KT}}]$, $\boldsymbol{l_j}$ is the j-th eigenvector and $\boldsymbol{A}$ is the diagonal matrix with corresponding eigen values $a_1 \geq \cdots \geq a_j \geq \cdots \geq a_{KT}$ organized in descending order, as stated in reference [Liu *et al.*, 2004; Huang *et al.*, 2002] and we observed the same phenomena that beyond certain index m, the trailing eigen values a_j where $m \leq j \leq KT$ are very small up until 0. For that reason, we decompose the eigenvector space into two sub-spaces as

$$\boldsymbol{L} = [\boldsymbol{L_s}, \boldsymbol{L_{null}}] \quad (4)$$

where $\boldsymbol{L_s} = [\boldsymbol{l_1}, \cdots, \boldsymbol{l_m}]$, $\boldsymbol{L_{null}} = [\boldsymbol{l_{m+1}}, \cdots, \boldsymbol{l_{KT}}]$, $\boldsymbol{L_{null}}$ forms the null space. It is reported in previous work [Liu *et al.*, 2004; Huang *et al.*, 2002] that common null space of the total covariance matrix does not contain useful information of the positive class samples and can thus be removed for more efficient computation. We then remove the common null space and transform the data from original feature space into eigen signal space, with the transformation matrix $\boldsymbol{L_s}$

$$\boldsymbol{q_{pi}} = \boldsymbol{L_s^T} \cdot \boldsymbol{x_{pi}}, \quad i = 1, \cdots, P \quad (5)$$

$$\boldsymbol{q_{nj}} = \boldsymbol{L_s^T} \cdot \boldsymbol{x_{nj}}, \quad j = 1, \cdots, N \quad (6)$$

All subsequent computation and sampling are performed in the eigen signal space for two reasons. First, the eigen signal space is typically much smaller in dimension than the original feature space and thus saves computation time. Second, it eliminates the possibility of drawing samples that have variance in the null space that can introduce undesired noise. After removing common null space and transforming original data into eigen signal space, we compute the covariance matrix for the minority class:

$$\boldsymbol{W_p} = \frac{1}{P}\sum_{i=1}^{P}(\boldsymbol{q_{pi}} - \overline{\boldsymbol{q}})(\boldsymbol{q_{pi}} - \overline{\boldsymbol{q}})^T \quad (7)$$

where $\overline{\boldsymbol{q}}$ is the positive class mean vector $\overline{\boldsymbol{q}} = \frac{1}{P}\sum_{i=1}^{P}\boldsymbol{q_{pi}}$ We then perform eigen decomposition just on the positive class covariance matrix:

$$\boldsymbol{D} = \boldsymbol{V^T} \cdot \boldsymbol{W_p} \cdot \boldsymbol{V} \quad (8)$$

where $\boldsymbol{D}$ is the diagonal matrix with the eigen values $d_1 \geq \cdots \geq d_j \geq \cdots \geq d_m$ organized in descending order, and $\boldsymbol{V^T}$ is the corresponding eigen vector matrix. Now we are able to create synthetic samples drawing from a multivariate Gaussian distribution given the positive class covariance matrix. However there is one remaining issue that needs to be addressed. For imbalanced time series data, it is often the case that the number of positive class samples is significantly smaller than the feature dimension, $P \ll KT$. The estimated covariance matrix tends to overfit to the few positive samples in the training set and thus does not generalize well to the test data. We conducted an experiment on one of the UCR multivariate time series benchmark datasets [Bagnall *et al.*, 2018]. First, we plot the eigen spectrum derived from the training data (see Figure 2(a) blue curve), then we project the test data onto the eigen vectors to measure the test data variance spectrum, represented in Figure 2(a) by the yellow curves with each one representing one sample in the test set. It can be clearly seen that the large eigen values do truthfully represent the variance in the test set while the smaller ones do not. We divide the eigen spectrum into reliable and unreliable regions and we regularize the unreliable region by adopting a smooth eigen spectrum model [Jiang *et al.*, 2008]. We compute the regularized eigen spectrum $\hat{d_j}$ as

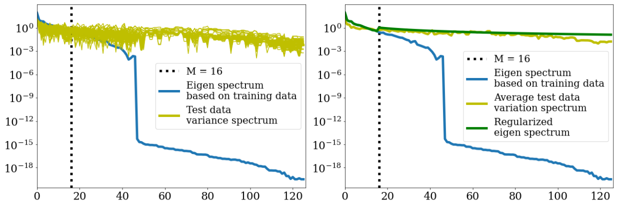
$$\hat{d_j} = \begin{cases} \frac{\rho}{j+\theta}, & \text{for } j \geq M \\ d_j, & \text{for } j < M \end{cases} \quad (9)$$

where $\hat{d_j} = \rho/(j + \theta)$ is the smooth eigen spectrum model in [Jiang *et al.*, 2008] and M is a hyperparameter representing the index where the eigen spectrum starts to depart from the test variance spectrum. Parameters ρ and θ are given by:

$$\rho = \frac{d_1 d_M (M-1)}{d_1 - d_M}, \quad \theta = \frac{M d_M - d_1}{d_1 - d_M} \quad (10)$$

The regularized eigen spectrum is represented by the smooth green curve in Figure 2(b) where we clearly observe that regularized eigen spectrum represents the test data variance spectrum much more accurately.

With regularized eigen spectrum, we draw synthetic minority samples $\boldsymbol{q_{pk}}$ from a multivariate Gaussian distribution given the regulated covariance matrix until we meet the target number of synthetic minority samples required. After initial synthetic samples $\boldsymbol{q_{pk}}$ are generated, we perform additional



(a) (b)

Figure 2: Eigen Spectrum Regularization. Part (a) shows the computed eigen spectrum based on training data and the test data variance spectrum. Part (b) shows the average spectrum of test data variance and the regulated eigen spectrum. It can be clearly seen that the regularized eigen spectrum do truthfully represent the average variance spectrum in the test data.

optimization step using the proposed SEMO algorithm detailed in Section 2.3. After the optimization with SEMO, we get the updated synthetic minority samples $\boldsymbol{q'_{pk}}$, which we map back to original feature space as $\boldsymbol{x'_{pk}}$ according to

$$\boldsymbol{x'_{pk}} = \boldsymbol{q'_{pk}} \times \boldsymbol{L_s^T},\ for\, k = P+1, \cdots, N \qquad (11)$$

to perform the classification task in the original feature space.

2.3 Optimization Algorithm

We first create the initial synthetic minority samples using the structure-preserving method described in Section 2.2, and then use the structure-preserving Expectation-Maximization Oversampling (SEMO) algorithm described next to simultaneously optimize the Gaussian mixture model parameters and update synthetic samples with those from the optimized distribution. It is worth noting that all computation in the optimization algorithm is performed in eigen signal space. Suppose we have M Gaussian mixture models in the minority class distribution, in order to re-establish class balance, the target number of synthetic minority samples is $N-P$. We start the algorithm from the original minority training data $\boldsymbol{q_{pi}}$ and initial synthetic minority samples $\boldsymbol{q_{pk}}$. The joint log likelihood of all minority class samples including the original and synthetic ones is given by

$$\ln P = \sum_{i=1}^{N} ln \sum_{k=1}^{M} \pi_k \mathcal{N}(q_{pi}|\hat{\mu_k}, \hat{\sigma_k^2}) \qquad (12)$$

Algorithm 1 Structure-Preserving EM Oversampling

```
**Input**: Training data $\boldsymbol{q_{pi}}$, Initial synthetic samples $\boldsymbol{q_{pk}}$
**Parameter**: $N-P$, $R$, $M$
**Output**: Optimized synthetic samples $\boldsymbol{q'_{pk}}$
**Initialize**: $\hat{\mu_k}$, $\hat{\sigma_k^2}$, $\hat{\pi_k}$ for $k = 1, \cdots, M$
**loop**
  1: Draw $\hat{\pi_k} \cdot (N-P) \cdot R$ samples, for $k = 1, \cdots, M$
  2: Replace $\hat{\pi_k} \cdot (N-P) \cdot R$ existing synthetic minority samples according to FIFO scheme
  3: Expectation step: Compute responsibilities $\gamma(z_{ik})$, $k = 1, \cdots, M$, $i = 1, \cdots, N_k$
  4: Maximization step: Update $\hat{\mu}\prime_k$, $\hat{\sigma}\prime_k$, $\hat{\pi}\prime_k$
**return** Optimized synthetic samples $\boldsymbol{q'_{pk}}$
```

At the beginning of each iteration, we first draw $\hat{\pi_k}(N-P)R$ samples from each estimated Gaussian model. Assuming the total number of synthetic samples we want to generate is $N-P$, here R is a hyperparameter denoting the percentage of total synthetic samples we want to replace in each iteration of the SEMO algorithm. We have experimented with different values of R and found that 0.05 is a good empirical value. We will then replace $\hat{\pi_k}(N-P)R$ samples in the synthetic samples pool with those from the latest optimized distribution in each of the M Gaussian model. After we replace the samples, we perform a standard expectation step where we compute the responsibility of each data sample given by:

$$\gamma(q_{pik}) = \frac{\hat{\pi_k} \mathcal{N}(q_{pi}|\hat{\mu_k}, \hat{\sigma_k^2})}{\sum_{j=1}^{M} \pi_j \mathcal{N}(q_{pi}|\hat{\mu_j}, \hat{\sigma_j^2})} \qquad (13)$$

where i is the sample index within each Gaussian Model and k is the index of Gaussian model $k = 1, \cdots, M$. Finally, we perform the maximization step by updating $\hat{\mu_k}\prime$, $\hat{\sigma_k^2}\prime$ and $\hat{\pi_k}\prime$ using the maximum likelihood estimator.

A summary of the optimization flow is given in Algorithm 1. Note that even though our optimization algorithm is based on the Expectation-Maximization framework, it differs from the standard Expectation-Maximization algorithm in the sense that the underlying samples are not fixed because we are replacing a portion of the synthetic samples at the beginning of each iteration. Because of this, we do not expect to see a strict increase in log likelihood after each iteration of an expectation-maximization step. For this reason the stopping criteria is set based on the classification performance measured by the F1-score on the validation set.

3 Experimental Study

In this section, we applied the proposed learning framework to various multivariate time series datasets and compared the classification performance between the proposed algorithm and existing benchmark oversampling algorithms. Following an introduction of datasets and benchmark algorithms used for the numerical experiments, we present a visual comparison of synthetic samples generated using different oversampling algorithms. Then, we present the classification performance comparison using the proposed algorithm and benchmark oversampling algorithms. Lastly, we present the results on classification performance improvement over iterations of the SEMO optimization algorithm.

3.1 Datasets and Benchmark Algorithms

Datasets from University of California Riverside (UCR) multivariate time series repository [Bagnall *et al.*, 2018] were used to conduct the numerical experiments. Table 1 contains detailed information for each dataset used including length of temporal and feature dimension, number of samples in majority ($\#Neg$) and minority ($\#Pos$) class and imbalance ratio.

Datasets	#Time Series	#Features	Train			Test	
			#Pos	#Neg	Imb.R.	#Pos	#Neg
RacketS.	30	6	11	108	0.10	43	109
FingerM.	50	28	16	157	0.10	49	51
NATOPS	51	24	10	150	0.07	30	150
UWaveG.	315	3	15	105	0.14	40	280
Libras	45	2	12	168	0.07	12	168

Table 1: Description of Multivariate Time Series data

For datasets that are presented in a multi-class format, we first convert them into a binary-class format by picking one label as the minority class and assigning the rest of the labels to the majority class. Note that some of the original datasets were balanced, for which we first downsample the minority class to create an imbalanced dataset.

Existing benchmark algorithms used for the comparison study include SMOTE [Chawla *et al.*, 2002] and ADASYN [He *et al.*, 2008]. The two benchmark algorithms are both interpolation-based oversampling algorithm. SMOTE is applied by first flattening the multivariate time series data in the same fashion our proposed algorithm handles it. Then the oversampling algorithm is applied to the flattened Multivariate time series data by interpolating between existing minority samples. Adaptive Synthetic Sampling (ADASYN) is applied in a similar fashion with the only difference in the algorithm itself that ADASYN adaptively determines the number of synthetic minority samples to generate based on the prevalence of neighboring majority samples. Other existing structure-preserving-based algorithms such as INOS [Cao *et al.*, 2013] is proposed for univariate time series data thus not directly applicable to the multivariate time series data used in our experimental study.

3.2 Visual Comparison

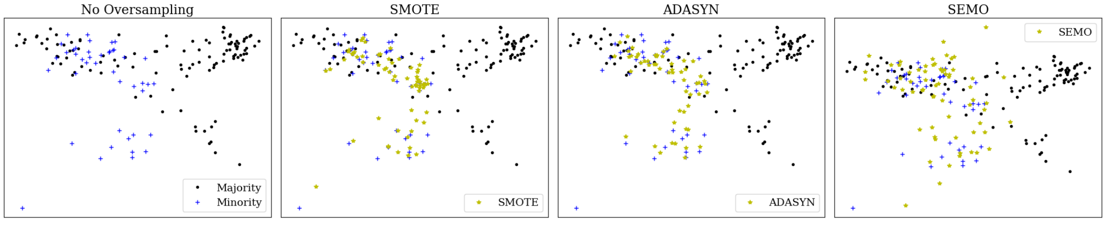
Figure 3 provides a visualization comparison of synthetic minority samples generated with different oversampling techniques. Part (a) shows the original majority and minority samples, with minority class samples represented by blue crosses and majority class samples represented by black dots. Part (b) to (d) show synthetic minority samples, represented by yellow stars, generated by SMOTE, ADASYN and our proposed SEMO algorithm, respectively. The X and Y axes correspond to the variance projections onto 1st and 2nd eigenvectors in the eigen signal space.

It can be seen that the interpolation-based oversampling methods, SMOTE and ADASYN, create synthetic minority samples that are closely associated with existing minority samples. This is determined by the way the algorithm generate synthetic samples, interpolating between existing minority samples. On the other hand, our proposed framework create synthetic samples that are not bounded by existing minority samples. It can be seen that samples created by the proposed framework conform more naturally to the original minority class data distribution. From another viewpoint, synthetic minority samples created with our framework can expand into the region where original minority samples do not exist. This is usually beneficial as test data are most likely not entirely within boundary of training samples, by creating synthetic samples that extends into the vicinity of training set, the classifier can learn a decision boundary that generalizes better on the test set.

3.3 Classification Performance Comparison

We performed the classification task on each of the datasets using the following three classification models: logistic regression, support vector machine, and random forest classification model. Every model was tested in conjunction with each different oversampling technique. The modeling pipeline, including generation of synthetic minority samples, model fitting and model evaluation, was repeated 10 times for each combination of classification model and oversampling algorithm. Average F1-score, precision and recall were recorded for each combination. Due to the unique nature of each dataset, the best-suited classification model can be different. For that reason we first pick the model that resulted in the best classification performance. Then, we list the performance using the same winning model combined with each oversampling algorithm. For the datasets RacketSports (RS) and NATOPS (NA), the winning model is logistic regression. For the datasets FingerMovements (FM), UWaveGestureLibrary (UWG) and Libras (LI), the winning model is random forest classification model.

Results are summarized in Table 2. Across the five datasets used in our experimental study, our proposed oversampling framework outperforms second-best methods for 4 out of 5 datasets and percentage improvement in terms of F1-score is in the range between 1.2% and 8.8%. Average F1-score improvement is 3.44%. The key performance improvement with the proposed framework is attributed to the excellent recall score. Across all datasets, 4 out of 5 demonstrated significant improvement in recall score. Percentage improvement of recall score is in the range between 7.5% and 85%. On average, recall score improves by 36.92%. Improvement on recall score is the key advantage of our proposed method due to the fact that the synthetic minority samples created by our proposed approach preserves the underlying correlations along the temporal and feature dimensions, thus enabling the classification model to learn the characteristics of minority sample better. From the visualization comparison, we can see that samples created with our proposed framework expands into the vicinity of the original minority sample distribution. In our observation, test data samples are usually present not only within boundaries of current minority samples in training set, but also outside the boundaries of training minority samples. By creating samples that are preserving the underlying covariance structure but not strictly associated with existing samples, the classifier can learn a more precise boundary that generalize better in the test set. It is also expected that as recall score improves, precision will drop due to the slight increase in false positives that happens with the significant increase in true positives. Overall classification performance is improved with our proposed method, as demonstrated by the improvement of F1-score.



(a) (b) (c) (d)

Figure 3: Visual Comparison of synthetic samples created by different oversampling techniques

F1-Score	RS	FM	NA	UWG	LI	Avg
No oversampling	0.333	0.167	0.636	0.644	0.500	0.456
SEMO	**0.557**	**0.248**	**0.752**	**0.843**	**0.914**	0.663
ADASYN	0.529	0.233	0.743	0.856	0.791	0.630
SMOTE	0.532	0.236	0.735	0.864	0.840	0.641
Improvement%	4.5	5.1	1.2	−2.4	8.8	3.44
Precision	RS	FM	NA	UWG	LI	Avg
No oversampling	0.818	0.455	1.000	1.000	1.000	0.855
SEMO	0.771	0.523	0.949	0.826	0.949	0.804
ADASYN	0.775	0.509	0.904	0.915	0.958	0.812
SMOTE	0.787	0.526	0.902	0.933	0.980	0.826
Improvement%	−5.7	−0.06	4.9	−11	−5.1	−3.39
Recall	RS	FM	NA	UWG	LI	Avg
No oversampling	0.209	0.102	0.133	0.475	0.333	0.250
SEMO	**0.437**	**0.163**	**0.617**	**0.860**	**0.883**	**0.592**
ADASYN	0.402	0.104	0.257	0.660	0.675	0.420
SMOTE	0.402	0.092	0.333	0.800	0.692	0.464
Improvement%	8.7	56.0	85.0	7.5	27.6	36.92

Table 2: Classification Performance Comparison in terms of average F1-score, Precision and Recall

3.4 Performance Improvement with SEMO

In this section, we present the result on classification performance improvement brought by the proposed Structure-Preserving Expectation-Maximization Oversampling (SEMO) algorithm. For each dataset, classification was performed using the synthetic minority data generated after each iteration of the SEMO algorithm. Classification performance was measured in terms of F1-score, precision and recall. For each dataset, the model pipeline, including synthetic minority sample generation, model fitting and model evaluation, was repeated 20 times and the average performance metrics were calculated. Figure 4 shows the delta of F1-score, precision and recall over iterations of the optimization algorithm. Part (a) and Part (b) correspond to two different datasets (Libras and UWaveGestureLibrary). It can be seen that classification performance first improves and then starts to decrease. It is understandable that classification performance increases as the synthetic minority samples used are from a better optimized distribution. The reason that the performance starts to decrease after a certain number of iterations appears to be that the distribution starts to overfit to the training samples and thus the generalization performance on the test set degrades. The number of iterations corresponding to the optimum classification performance is a hyperparameter that needs to be configured. Based on the plot shown in Figure 4 and our experimental study on other datasets, the best iteration number appears to be in the range between 2 and 4.

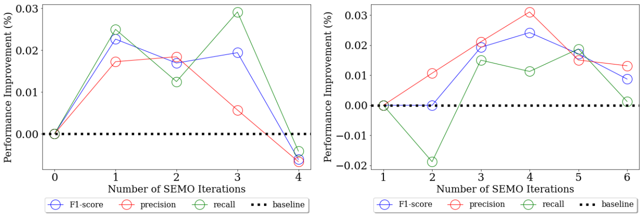


Figure 4: Performance improvement from SEMO optimization

4 Conclusion

In this paper, a learning framework is proposed to create synthetic minority samples for imbalanced multivariate time series data. The proposed framework models the minority class as a Gaussian mixture model and creates the initial synthetic minority samples via a structure-preserving approach. Then, the SEMO algorithm jointly optimizes the minority class distribution and updates the synthetic minority samples with those from an optimized distribution. Numerical experiment results showed that the proposed learning framework can achieve a better classification performance in terms of average F1 score by 3.44% and an improvement in recall score by 36.92%.

The proposed method and results are particularly useful given the fact that many real-world time series classification tasks involve multivariate time series. To our best knowledge, the proposed framework is the first oversampling framework that directly addresses this particular task.

References

[Bagnall *et al.*, 2018] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018, 2018.

[Batista *et al.*, 2004] Gustavo E. A. P. A. Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 2004.

[Cao *et al.*, 2011] Hong Cao, Xiao-Li Li, Yew-Kwong Woon, and See-Kiong Ng. Spo: Structure preserving oversampling for imbalanced time series classification. In *IEEE 11th International Conference on Data Mining*, pages 1008–1013, Vancouver,BC, 2011. IEEE.

[Cao *et al.*, 2013] Hong Cao, Xiao-Li Li, David Yew-Kwong Woon, and See-Kiong Ng. Integrated oversampling for imbalanced time series classification. *IEEE Transactions on Knowledge and Data Engineering*, pages 2809–2822, 2013.

[Cao *et al.*, 2014] Hong Cao, Vincent Y. F. Tan, and John Z. F. Pang. A parsimonious mixture of gaussian trees model for oversampling in imbalanced and multimodal time-series classification. *IEEE Transactions on Neural Networks and Learning Systems*, pages 2226–2239, 2014.

[Chawla *et al.*, 2002] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, pages 321–357, 2002.

[Chawla *et al.*, 2003] Nitesh V. Chawla, Aleksandar Lazarevic, Lawrence O. Hall, and Kevin W. Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In *Proceedings of European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 107–119, Berlin, Heidelberg, 2003. Springer.

[Chen *et al.*, 2010] Sheng Chen, Haibo He, and Edwardo A. Garcia. Ramoboost: Ranked minority oversampling in boosting. 2010.

[Estabrooks *et al.*, 2004] Andrew Estabrooks, Duke Taeho Jo, and Nathalie Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 2004.

[Guo and Viktor, 2004] Hongyu Guo and Herna Lydia Viktor. Learning from imbalanced data sets with boosting and data generation: The databoost-im approach. 2004.

[Han *et al.*, 2005] Hui Han, Wen-Yuan Wang, , and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Proceedings of International Conference on Intelligent Computing*, pages 878–887, Berlin, Heidelberg, 2005. Springer.

[He and Garcia, 2009] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[He *et al.*, 2008] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of 2008 IEEE International Joint Conference on Neural Networks*, pages 1322–1328, Hong Kong, China, June 2008. IEEE.

[Huang *et al.*, 2002] Rui Huang, Qingshan Liu, Hanqing Lu, and Songde Ma. Solving the small sample size problem of lda. In *Object recognition supported by user interaction for service robots*, pages 29–32, Quebec City, Quebec, Canada, 2002. IEEE.

[Jiang *et al.*, 2008] Xudong Jiang, Bappaditya Mandal, and Alex Kot. Eigenfeature regularization and extraction in face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[Liu *et al.*, 2004] Wei Liu, Yunhong Wang, S.Z. Li, and Tieniu Tan. Null space-based kernel fisher discriminant analysis for face recognition. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 369–374, Seoul, South Korea, 2004. IEEE.

[Liu *et al.*, 2009] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2009.

[Runger, 2015] Mustafa Gokce Baydogan George Runger. Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery*, 2015.

[Sun *et al.*, 2007] Yanmin Sun, Mohamed S. Kamel, Andrew K.C. Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 2007.

[Wei and Keogh, 2006] Li Wei and Eamonn Keogh. Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 748–753, Rome, Italy, August 2006. Morgan Kaufmann.