Learning Multivariate Time Series with Unsupervised Representation Methods using Simulated Datasets

Thabang Lebese, Cécile Mattrand, David Clair, Jean-Marc Bourinet

Learning Multivariate Time Series with Unsupervised Representation Methods using Simulated Datasets

T. Lebese[1]; C. Mattrand[2]; D. Clair[3]; JM. Bourinet[4]

[1,4]LIMOS, Université Clermont Auvergne, France; [2,3,4]SIGMA Clermont, [2,3]Institut Pascal, Université Clermont Auvergne, France

International Symposium on Intelligent Data Analysis (IDA 2023)

1. Overview

- **Unsupervised representation learning** is useful for extracting representations from unlabeled multivariate time-series
- Extracted representations can match **underlying patterns or states**
- We explore **3 unsupervised** learning SOTAs: Triplet Loss, TNC and CPC
- We evaluate each on three **simulated datasets**
- Evaluations are on two downstream tasks: **clustering** and **classification**

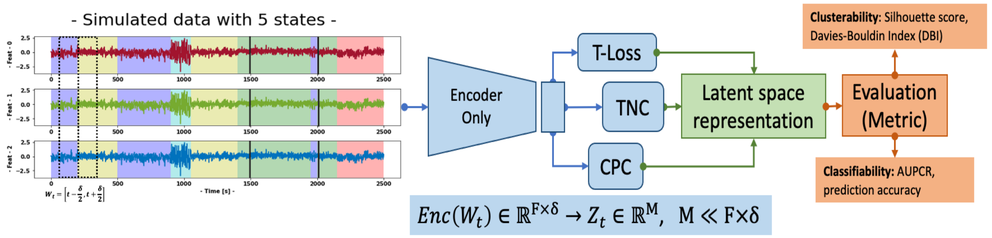
2. Motivation

- Multivariate time series are mostly unlabeled, potentially long and of unequal lengths in the same dataset
- Modeling MTS requires significant time and expertise
- Unsupervised representation learning helps with understanding of generating processes and boosts performance of subsequent ML tasks
- Objectives if this work:
  1) Implement and reproduce three unsupervised learning SOTAs (Triplet Loss[1], TNC[2] and CPC[3]**)** techniques on three simulated datasets
  2) Demonstrate their strengths, weaknesses and determine suitability of each
- We demonstrate their practicality in various scenarios

3. Baselines & training setup

- All baselines use a Deep **BiRNN** as encoder backbone
  - CPC uses **Gated Recurrent Units (GRU)** as a regressor
- Across baselines:
  - All **latent space dimensionality** set to 10
  - **Window size** set to 50 across baselines
  - We perform a 60/20/20 - **train/validation/test** data split accordingly
- Maximum **training epochs** limited to (140, 400, 500) for (TNC, CPC, Triplet loss)
- Cross baseline **hyper-parameter** tuning:
  - Window size selection
  - PU learning weight parameter (TNC)
  - No parameter tuning for the latent space

4. Methods



5. Datasets

TEN states									simulation 3	
FIVE states								simulation 2		
FOUR states							simulation 1			
Features	State 0	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9
Feature 0	GP(periodic)	$NARMA_\alpha$	GP(Sq.Exp)	$NARMA_\beta$	GP(Matern)	GP(Matern)	$NARMA_\beta$	GP(Sq.Exp)	$NARMA_\alpha$	GP(periodic)
Feature 1	GP(periodic)	$NARMA_\alpha$	GP(Sq.Exp)	$NARMA_\beta$	GP(Matern)	GP(Matern)	$NARMA_\beta$	GP(Sq.Exp)	$NARMA_\alpha$	GP(periodic)
Feature 2	GP(Sq.Exp)	$NARMA_\beta$	GP(periodic)	$NARMA_\alpha$	GP(Matern)	GP(Matern)	$NARMA_\alpha$	GP(periodic)	$NARMA_\beta$	GP(Sq.Exp)

Table I: Signal distributions for three simulated trivariate time series. Simulation 1 covers states 0-3, simulation 2 covers 0-4, and simulation 3 covers 0-9.

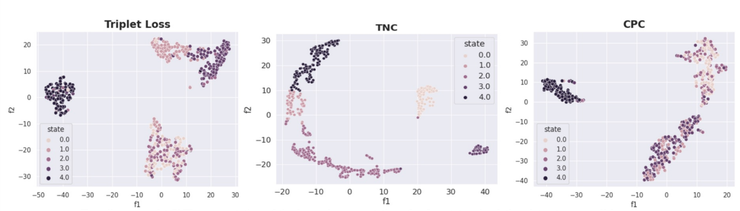
6. Results

		Triplet Loss				TNC				CPC			
	δ	AUPCR	Accuracy	Silhouette	DBI	AUPCR	Accuracy	Silhouette	DBI	AUPCR	Accuracy	Silhouette	DBI
Sim. 1	50	0.920	73.104	0.477	0.804	**0.998**	**97.167**	**0.750**	**0.330**	0.871	57.400	0.504	0.795
Sim. 2	50	-	82.033	0.484	0.866	-	**97.517**	**0.636**	**0.616**	-	65.183	0.492	0.757
Sim. 3	50	0.790	75.470	0.426	1.130	**0.949**	**90.180**	**0.648**	**0.509**	0.432	43.290	0.356	0.903
Avg.		0.855	76.869	0.462	0.933	**0.9735**	**94.9547**	**0.678**	**0.485**	0.6515	55.291	0.451	0.818

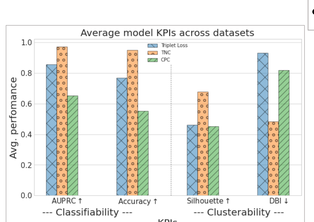
Table II: Baseline performance for classification (accuracy and AUPRC) and clustering (Silhouette and DBI) across all datasets.

Our evaluation metrics: KPIs

- **Clusterability:** TNC outperforms other two models
- These scores are evaluated on top of $k-means$[4] clusters
  - On Silhouette score and DBI: TNC has best overall scores
    - Overall performance decreases with dataset complexity and length
    - Triplet Loss has a better Silhouette score than CPC, but CPC has better DBI than Triplet Loss
- **Classifiability:** TNC outperforms other two models
  - We use AUPCR and Accuracy: overall best scores are with TNC
    - Overall performance diminishes with dataset complexity and length
    - Although Triplet Loss has better scores for both AUPCR and accuracy over CPC



Two-dimensional t-SNE representation of the learned representations for five states simulation dataset across all baselines.



- **Triplet Loss**[1] uses contrastive learning by sampling a tuple of triplets $(x^{pos}, x^{ref}, x_k^{neg})$
- There is weight sharing between similar (x^{pos}, x^{ref}) and non-similar (x_k^{neg}, x^{ref}) time series samples
- We sample multiple negative $x_{k\in[1,K]}^{neg}$ randomly chosen independently
- We optimize the objective:

$$\mathcal{L} = -\log\sigma\left(f_\theta(x^{ref})^T f_\theta(x^{pos})\right) - \frac{1}{K}\sum_{k=1}^{K}\log\sigma\left(-f_\theta(x^{ref})^T f_\theta(x_k^{neg})\right)$$

- **TNC**[2] combines contrastive learning with Positive Unlabeled (PU) Learning
- It has 3 components:
  1) Temporal Neighborhood: determining regions with similar underlying states
  2) Encoder: a sequential deep/shallow neural network
  3) Discriminator: a multi-headed binary classifier
- TNC optimizes the objective:

$$\mathcal{L} = -\mathbb{E}_{W_t\sim X}\left[\mathbb{E}_{W_t\sim N_t}[\log(\mathcal{D}(Z_t, Z))] + \mathbb{E}_{W_k\sim \bar{N}_t}[w_t\log(\mathcal{D}(Z_t, Z)) + (1-w_t)\log(1-\mathcal{D}(Z_t, Z))]\right]$$

- **CPC**[3] combines predictive coding with contrastive learning.
- The objective is to minimize mutual information between x_{t+1} and contextual latent states c_t
- Here components are jointly optimized via an InfoNCE loss
  1) Encoder: a sequential deep/shallow neural network
  2) Auto-regressor: constructing contexts from encodings
- CPC Optimizes the objective: $\mathcal{L} = -\mathbb{E}_X\left[\log\frac{f_k(x_{t+k}, c_t)}{\sum_{x_j\in X} f_k(x_j, c_t)}\right]$

7. Conclusion

- **TNC** outperformed two other methods on investigated KPI scores
- Weight parameter in TNC is important for the loss and representation quality
- Window size selection is crucial for performance across baselines
- TNC is a better option and worthy of further consideration

8. Future work

- Evaluate TNC on real MVT car data with 2.5 years of driving
- Explore scenarios such as increasing number of features to incorporate different sensory data
- Combine all strategies to learn larger datasets of multiple vehicles

References

1. Representation with Triplet loss [J.Y. Franceschi, et.al. (2020)]
2. Representation Temporal Neighborhood Coding [S. Tonekaboni, et.al. (2021)]
3. Representation with Contrastive Predictive Coding [A. van den Oord, et.al. (2019)]
4. K-means: Classification and analysis of multivariate observations [J. MacQueen, (1967)]
5. t-SNE: Visualizing data using t-SNE [L. Van der Maaten, et.al. (2008)]

Contact info: Thabang LEBESE (Thabang.Lebese@sigma-clermont.fr)