Frouros: A Python library for drift detection in Machine Learning problems

Jaime Céspedes Sisniega (cespedes@ifca.unican.es)
Instituto de Física de Cantabria (IFCA), CSIC-UC

Summary

- Frouros is a Python library capable of detecting drift in machine learning problems by providing a combination of classical and more recent algorithms, both for *concept drift* and *data drift*.
- Two main objectives: 1. to easily integrate drift detection methods with the use of the *scikit-learn* library; 2. to unify in a single library *concept drift* and *data drift* methods.

Background

- Deploying machine learning models in real-world applications often assumes that the same concepts learned during the traning phase will still be valid at inference time.
- Due the high cost of collecting and labeling samples, many deployed machine learning models, can only rely on distribution changes.

  We have adopted the following definitions for the types of drift that can occur:
- *Concept drift*: There is a change in the conditional probability P(y|X), with or without a change in P(X). Also known as *real concept drift* [1].
- *Virtual drift:* There is a change in P(X) but does not affect conditional probability P(y|X).
- *Data drift*: As well as *virtual drift*, there is a change in P(X) but due the fact that there is no labeled data y available, it cannot be verified that P(y|X) is being affected or not. Therefore, this type of drift only focuses in the distribution of the covariates P(X) [2].

Supervised methods

- These methods try to detect *concept drift*, so in order to update the detector they require ground-truth labels of the predictions that have been previously made.
- Frouros offers 12 supervised methods that can be classified in DDM (Drift Detection Method) Based, CUSUM (Cumulative Sum) Based and Window Based.
- The detector wraps the *scikit-learn* estimator and receives through the update method the value to update detector's statistics and check if drift is occurring on each iteration.
- Frouros provides the following helper classes, called modes, to interact with the detector's update process: Incremental Learning Mode and Normal Mode.

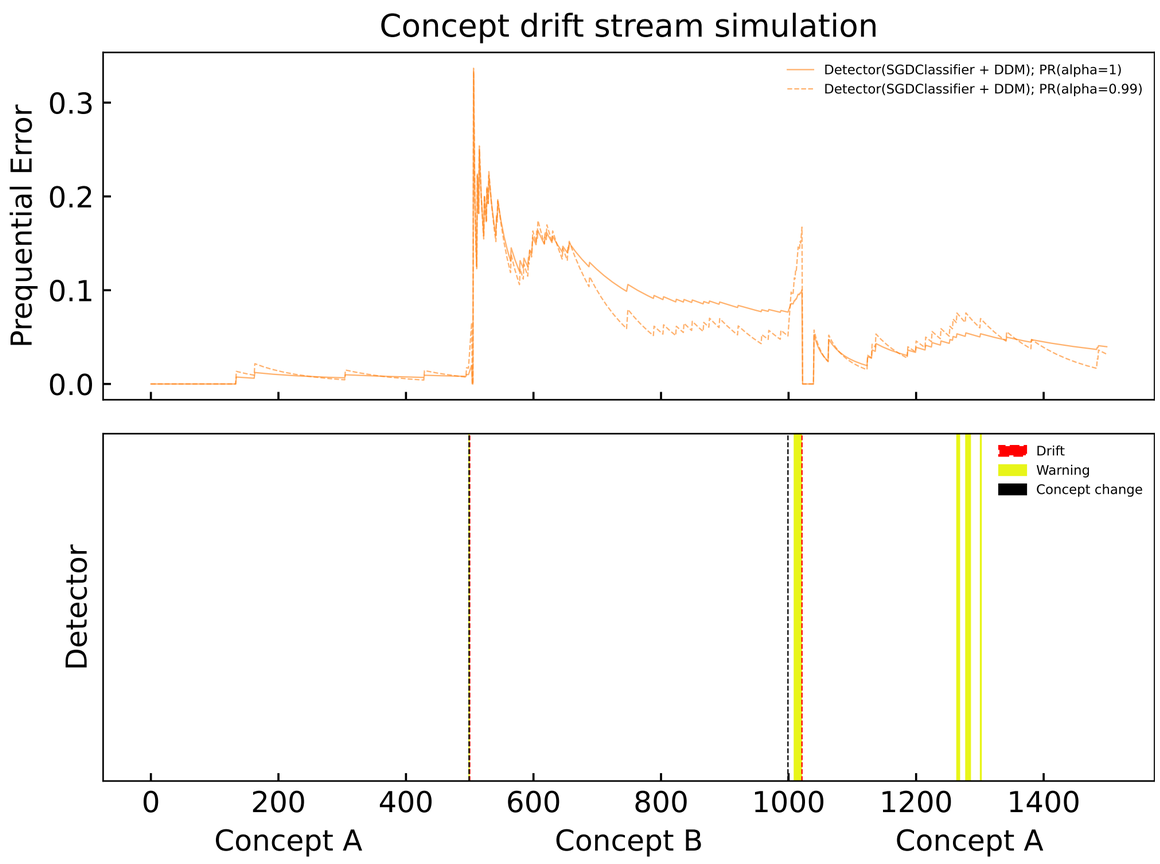

Figure 1: Example of using Frouros with the supervised method DDM (Drift detection Method) [3]. SGDClassifier is used in combination with DDM detector. Trained beforehand on the Concept A and tested and trained with an incremental learning mode on Concept A [0, 499], Concept B [500, 999] and again on Concept A [1000, 1499].

Unsupervised methods

- These methods try to detect *data drift* by considering only the covariates regardless of the existence or not of labels.
- Frouros offers 10 unsupervised methods that can be classified in distance based and statistical test methods.
- *fit* method stores the reference distributions and *transform* method applies the corresponding algorithm to compare new samples distribution with the reference distribution.
- Depending on the type of method used, *distance* (distance based) and *test* (statistical test) attributes can be acceded after calling *fit* and *transform* methods.

```python
import numpy as np
from sklearn.gaussian_process.kernels import RBF
from frouros.unsupervised.distance_based import MMD

np.random.seed(seed=31)
# Generate reference and test distributions
X_ref = np.random.multivariate_normal(mean=np.ones(2), cov=2*np.eye(2), size=100)
X_test = np.random.multivariate_normal(mean=np.zeros(2), cov=np.eye(2) + 1, size=100)
# Set a significance level for the hypothesis test
alpha = 0.01
# Define, fit and transform the detector
detector = MMD(num_permutations=1000, kernel=RBF(length_scale=1.0), random_state=31)
detector.fit(X=X_ref)
detector.transform(X=X_test)
# Obtain the p-value and perform the hypothesis test
mmd, p_value = detector.distance
p_value < alpha
>>> True  # Data drift detected. Both samples come from different distributions.
```

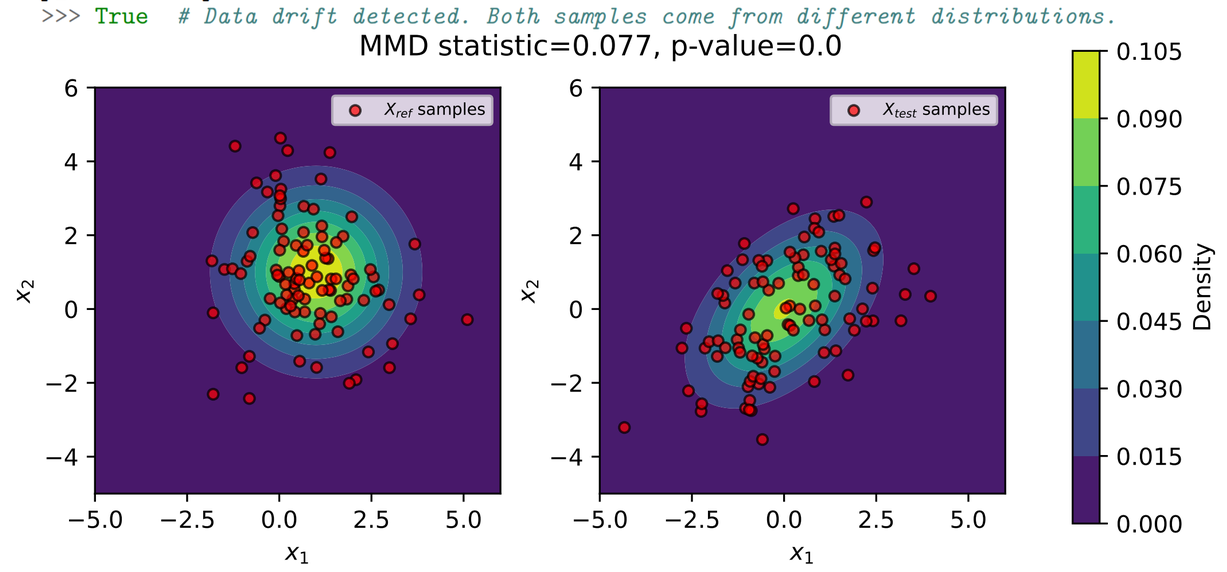

Figure 2: Example of using Frouros with the unsupervised method MMD (Maximum Mean Discrepancy) [4]. Two bivariate normal distributions are used to verify with MMD that both generate samples come from different distributions.

Future work

- Planning to adapt some supervised methods to support *batch-incremental learning* as long as the nature of each algorithm allows it.
- Extend the unsupervised part to include methods that work with individual instances and not only in batch mode.
- Adding new helper classes that interact with the detector would make it possible to adapt the library to handle more real-world use cases.

Development

- Source code repository: github.com/IFCA/frouros
- Documentation API: frouros.readthedocs.io
- Python Package Index (PyPI): pypi.org/project/frouros/
- Preprint repository: arXiv:2208.06868
- License: BSD-3-Clause

Acknowledgements

The authors acknowledge the funding from the Agencia Estatal de Investigación, Unidad de Excelencia María de Maeztu, ref. MDM-2017-0765.

References

[1] João Gama,Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

[2] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. *In Brazilian symposium on artificial intelligence*, pages 286–295. Springer, 2004.

[4] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.