Evaluating Data Distribution Based Concept Drift Detectors

Author:
Konsta Kanniainen
K.O.Kanniainen@student.tudelft.nl

Responsible professor:
Jan Rellermeyer
rellermeyer@vss.uni-hannover.de

Supervisor:
Lorena Poenaru-Olaru
L.Poenaru-Olaru@tudelft.nl

Detecting Concept Drift

- Deployed machine learning models lose accuracy over time as the input data evolves - this is called concept drift. [1]
- Labels are not always easy to obtain for evaluation in production settings.
- Label-independent drift detectors have been researched [2], but code is rarely available for reproduction and further studies.
- Goal: Implement and evaluate data distribution based concept drift detectors.

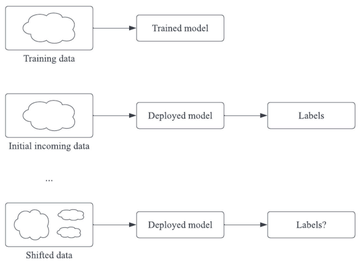


Figure 1. Problem illustration. Image: Own work

Experiment Setup

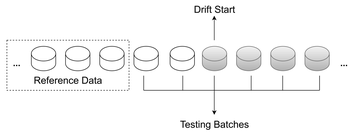


Figure 2. Experiment setup with synthetic data. Image: [1]

- Data stream containing concept drift(s) is divided into reference (training) data and testing batches.
- For synthetic datasets, the batch with a drift is known and drift sizes between 500 and 20000 rows are used for gradual drift; for real-world datasets, the drifts are determined by model accuracy drop.
- Metrics for synthetic data:
  1) False Positive Rate (FPR_s) = *# alerts before drift / # batches before drift*
  2) Latency = *delay from drift to alert / batches after drift*
- Metrics for real-world data:
  1) False Positive Rate (FPR_rw) = *# false alarms / # total non-drifting batches*
  2) Accuracy = *# correctly detected batches / # total drifting batches*

Implementations

SyncStream [3]

Two drift detection techniques are included in the algorithm, both targeting abrupt drift. Because labels are considered to be unavailable in this study, the techniques are used on the whole batch of incoming data instead of label-wise collections.

1) Principal component analysis (PCA):
- Eigenvectors are computed by decomposing the covariance matrix (sklearn.decomposition.PCA)
- Angle between consecutive batches is determined and 30° used as threshold (own implementation)

2) Statistical test (Wilcoxon rank sum test):
- Brunner and Munzel's generalized Wilcoxon test statistic is computed for consecutive batches (scipy.stats.rankdata, own implementation)
- This follows standard normal distribution: p = 0.01 is used as threshold

SCD (Density test) [4]

This technique makes use of a kernel density estimator (KDE) trained on half of the reference data. Concept drift is measured as the weighted log-likelihood difference between the remaining half and the testing batch. This follows a normal distribution; p = 0.08 is used to limit false positives. The test may be conducted in two directions to increase power.

- The Expectation Maximization procedure to learn optimal bandwidths for the KDE was abandoned due to its high computation costs; the Scott method was used instead (scipy.stats.gaussian_kde)
- Variance estimation is done by bootstrapping, adapted to stop after 30 iterations instead of 4000 if the distribution had stabilized (own implementation)
- PCA preprocessing was added for some datasets to address linear algebra errors; this also turned out to improve drift detection performance

References

[1] L. Poenaru-Olaru, L. Cruz, A. van Deursen, and J. S. Rellermeyer, "Are concept drift detectors reliable alarming systems? - a comparative study," in *7th Workshop on Real-time Stream Analytics, Stream Mining, CER/CEP & Stream Data Management in Big Data*, 2022.

[2] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.

[3] J. Shao, Z. Ahmadi, and S. Kramer, "Prototype-based learning on concept-drifting data streams," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, (New York, NY, USA), pp. 412–421, Association for Computing Machinery, 2014.

[4] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Statistical change detection for multi-dimensional data," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, (New York, NY, USA), pp. 667–676, Association for Computing Machinery, 2007.

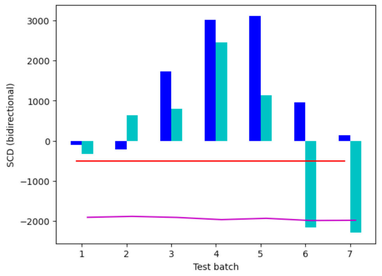
Results

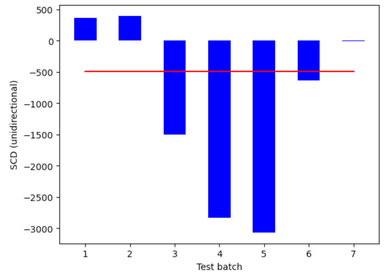
Perfect score
Potentially usable
Not usable

	Synthetic (FPR_s, Latency)			*Real-world (FPR_rw, Accuracy)*			
	SEA *(abrupt)*	AGRAW1 *(abrupt)*	AGRAW2 *(abrupt)*	Airlines	Elect2	Weather	Spam
SyncStream-PCA	(0.0, 0.25)	(0.0, 1.0)	(0.0, 1.0)	(0.0, 0.0)	(0.48, 0.54)	(0.90, 0.96)	(1.0, 1.0)
SyncStream-Stat	(0.0, 0.0)	(0.0, 1.0)	(0.0, 0.25)	**(0.0, 0.76)**	(1.0, 0.93)	(0.53, 0.61)	(0.88, 0.96)
SCD (Density test)	**(0.0, 0.0)**	**(0.0, 0.0)**	**(0.0, 0.0)**	-	(1.0, 1.0)	(0.37, 0.52)	**(0.20, 0.63)***

***Inverted test statistic**

- The best result for each combination is displayed.
- Performance under gradual drift did not significantly differ from abrupt drift.
- For these results, PCA preprocessing was used with SCD on the AGRAWs, Elect2, and Spam; the latter with dimensions reduced to 100.
- SCD could not run on Airlines due to the size of the dataset.
- SCD behavior changed significantly after adding PCA preprocessing:





Conclusions and Future Work

- Performance on synthetic data is no guarantee for performance on real-world data.
- PCA preprocessing should be investigated more, in addition to encoders and scalers.
- SyncStream-PCA does not seem to perform expectedly when used as a standalone drift detection method; the Wilcoxon test achieves some more convincing results.
- SCD has a very high runtime complexity compared to the SyncStream methods.
- The test statistic of SCD seems inverted with some configurations, which semantically means that the drifted distribution would be closer to the original than the original itself.
- From a ML model quality point of view, drift of the data distribution may not explain all of the drift; the decision boundary may also drift by itself [2]. This may limit the effectiveness of these methods.

TU Delft

Paper

Code