An Empirical Analysis of Data Drift Detection Techniques in Machine Learning Systems

Lucas Helfstein Rocha and Kelly Rosa Braghetto

IME-USP

2024

Introduction

- Machine learning models often process **data streams** without real-time feedback.
- This poses a challenge in maintaining **accuracy** and **robustness**.
- **Data drift detection** helps monitor input data and compares it to the data used during training (Lu et al. 2018).
- It also ensures consistency and prevents model degradation.

Introduction

- This work is focused on applying **data drift detection** techniques to enhance *classifier* performance employing **nonparametric methods**.
- Integration of drift detection into the classification pipeline allows:
  - **Dynamic adaptation** to changing data environments.
  - Improved model performance over time.

Types of drift

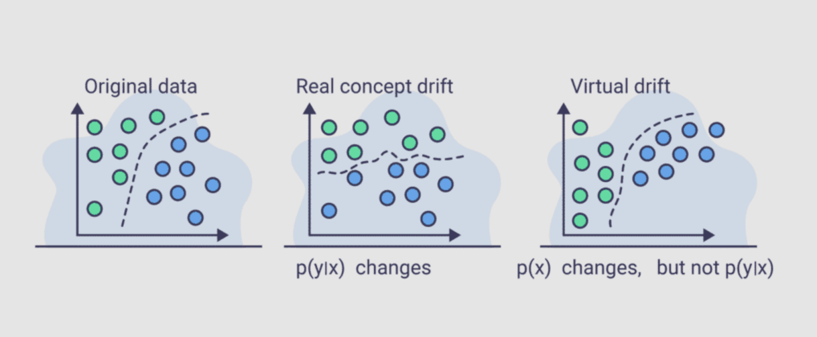


Image source: https://www.aporia.com/

Drift detection methods

- To detect drifts, it is essential to monitor the extent of divergence between the distributions of the data sets.
- **Distance-based methods**: provide a direct measure of the distance or dissimilarity between two probability distributions.
- **Statistical methods**: provide a statistical measure (e.g., p-value) indicating the likelihood that the two distributions are the same.

Kolmogorov-Smirnov Test

- The Kolmogorov-Smirnov (KS) **statistical test** can be used to test whether two samples came from the same distribution.
- Testing multiple hypotheses increases the probability of observing rare events, which elevates the likelihood of incorrectly rejecting a null hypothesis.
- To mitigate this, the **Bonferroni correction** (Bland and Altman 1995) can be used, it adjusts the significance level for multiple comparisons (Rabanser, Günnemann, and Lipton 2019).

Multiple Kolmogorov-Smirnov Tests

- For d feature distributions, the decision rule is to reject the null hypothesis at significance level α if:

$$\min_{k=1,2,\ldots,d} KS(F_k, G_k) > c\left(\frac{\alpha}{d}\right)\sqrt{\frac{n+m}{n \times m}}$$

- Where:
  - $KS(F_k, G_k)$ is the KS Test statistic for the empirical distribution functions F and G of the k-th dimension.
  - n and m are the respective sample sizes for the two distributions.
- The Bonferroni correction is applied by testing at significance level $\frac{\alpha}{d}$.

Distances

For two discrete probability distributions P and Q:

- The Kullback–Leibler (KL) Divergence:

$$KL(P||Q) = \sum_{x \in \mathcal{X}} P(x) log(\frac{P(x)}{Q(x)}) \tag{1}$$

- The Jensen–Shannon (JS) Divergence:

$$JS(P||Q) = \frac{1}{2} KL(P||\frac{(P+Q)}{2}) + \frac{1}{2} KL(Q||\frac{(P+Q)}{2}) \tag{2}$$

- The Hellinger distance $H(P,Q)$ is defined as:

$$H(P,Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{n} (\sqrt{p_i} - \sqrt{q_i})^2} \tag{3}$$

Drift Detection Method (DDM)

Inspired by (Ditzler and Polikar 2011):

- DDM assumes that data arrives in batches
- First batch will be a reference dataset
- For each new batch, the chosen distance is measured
- A delta between the measures is updated, and is used to update an adaptive threshold
- If the delta is bigger than the accepted threshold, reference dataset is set to be this new batch
- Else, the new batch gets added to the reference dataset

Drift Detection Method (DDM)

The distances for DDM used in this work were:

- HDDDM uses Hellinger Distance
- JSDDM uses Jensen-Shannon
- KSDDM uses Kolmogorov Smirnov, but without the adaptive threshold

Drift Detection Methods

- In terms of **computational cost**, the previous techniques are comparable.
- Each method derives empirical distributions from the same input data.
- Bins are separated in the same way.
- The computation of drift is:
  - Linear with respect to the number of bins.
  - The number of bins is dictated by the batch size.

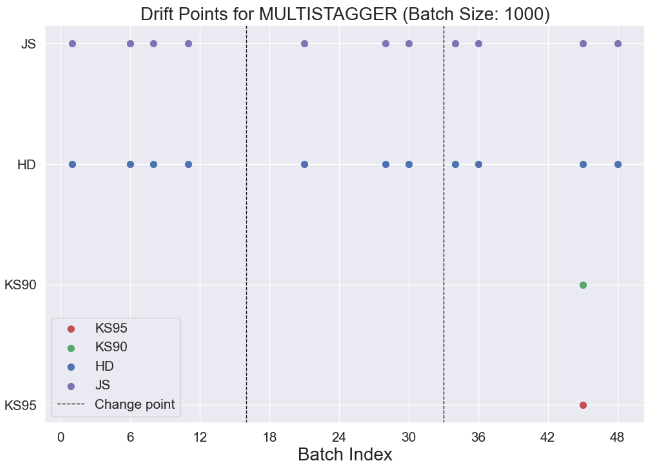
Datasets

- Insects
- SEA Datasets
- STAGGER Datasets
- Electricity
- Magic Gamma Telescope

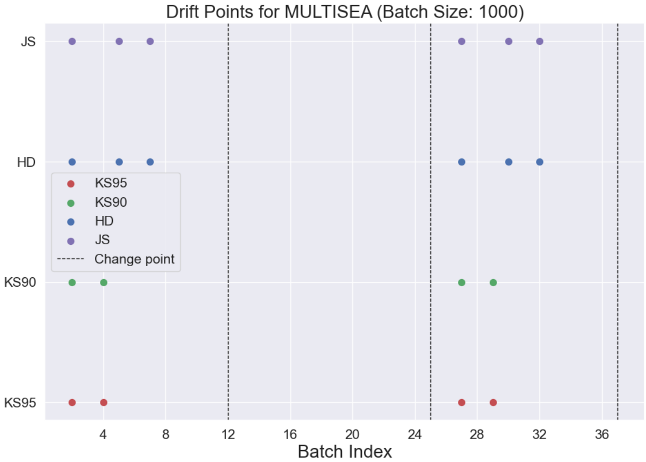
Using Drift Detection to Improve ML System's Performance

- Datasets were segmented into batches of 1000, 1500, 2000, and 2500 instances to balance evaluation and interpretability.
- These sizes were chosen to ensure a sufficient number of batches for drift detection techniques while avoiding complexity in visualization for larger datasets.
- The segmentation aimed to demonstrate each technique's sensitivity to varying batch sizes.

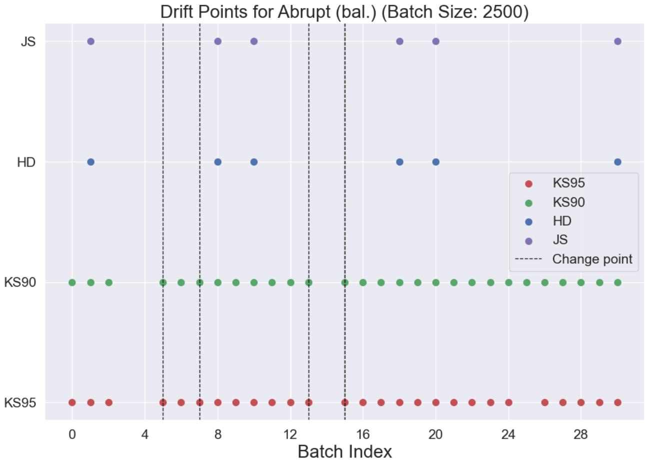
Detected Data Drifts - MULTISTAGGER 1000



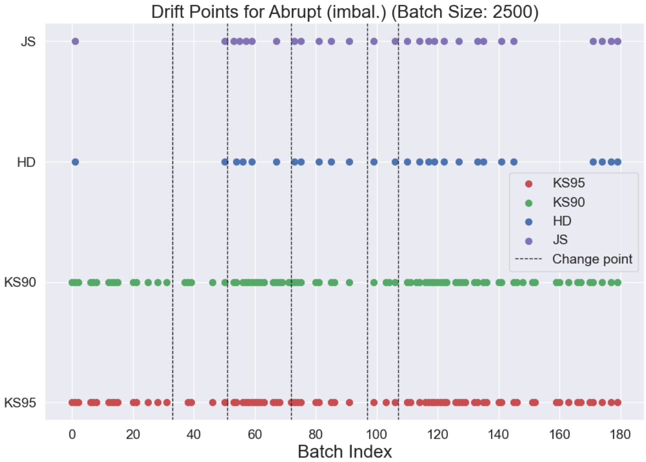
Detected Data Drifts - MULTISEA 1000



Detected Data Drifts - Abrupt bal 2500



Detected Data Drifts - Abrupt imbal 2500



An Approach for Using Detected Drifts to Improve a Classifier

1. **Input**: Labeled data batches and a drift detection technique.
2. Use batch 1 to train a Naive Bayes classifier C_B – the baseline model.
3. Use batch 1 to train a Naive Bayes classifier C_D – the model that benefits from drift detection.
4. Set batch 1 as the reference batch.

An Approach for Using Detected Drifts to Improve a Classifier

5. **From batch 2 onwards:**
   1. Store the predictions of both classifiers C_B and C_D for the current batch.
   2. Check for drift between the reference set and the current batch using the drift detection technique.
   3. Update C_B classifier with the current batch.
   4. **If no drift is detected:**
      - Update C_D classifier with the current batch.
      - Update the reference set by merging it with the current batch.
   5. **If drift is detected:**
      - Set the reference set to the current batch.
      - Reset classifier C_D training only with the new reference set.
6. **At the end of all batches:** Compute the performance metrics.

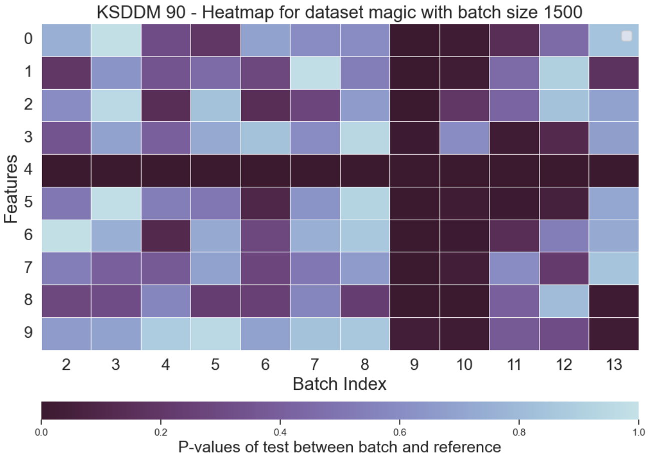
Experimental results

- The algorithm from the previous slides was implemented to compare the following techniques:
  - **Base**: No drift detection.
  - **KS95**: KSDDM with 95% of confidence.
  - **KS90**: KSDDM with 90% of confidence.
  - **HDDDM**: With the standard parameters (Ditzler and Polikar 2011).
  - **JSDDM**: With the standard parameters of HDDDM.
- Techniques were evaluated using datasets from slide 12, measuring various model metrics.
- The most suitable metrics for assessment were the **Area Under the Curve (AUC)** and the **F1 score**.

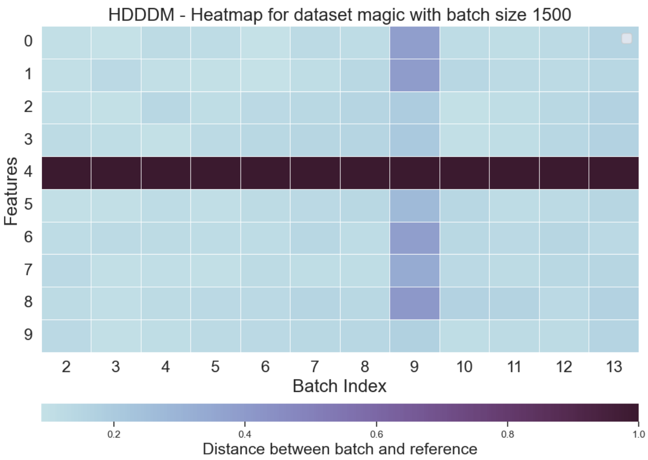
Experimental results

- Utilizing drift detection techniques for optimal retraining times significantly enhances overall performance.
- Smaller batches yielded the best results for the F1 and AUC metrics.
- In terms of detected drifts:
  - **KS90** triggered the most resets, closely followed by **KS95**.
  - **HDDDM** and **JSDDM** had similar results, triggering significantly fewer resets than the KS techniques.

Experimental results - KSDDM90



Experimental results - HDDDM



Experimental results

- The **KS Test** is highly sensitive to data shifts (see Section 22).
- This excessive drift detection can lead to overfitting, resulting in good F1 and AUC scores but low generalization to new data.
- In contrast, the adaptive thresholds of **HDDDM** and **JSDDM** enhance sensitivity over time:
  - Particularly beneficial for datasets with stable distributions, such as **MULTISTAGGER**.

Experimental results

- The detected drifts and full retraining with **HDDDM** and **JSDDM** led to better F1 and AUC metrics compared to the **KSDDM** method.
- Experimental results indicate that the analyzed drift detection techniques enhance system robustness, even in scenarios with concept drift.
- The approach of resetting the classifier improved performance in the presence of drifts, even with prequential evaluation.

Related work

- (Dasu et al. 2006) proposed a method using **KL Divergence** with an empirical evaluation on both real and synthetic data, showcasing the accuracy of this approach.
- (Pérez-Cruz 2008) proposed a method for estimating **KL Divergence** between continuous densities using the empirical cumulative distribution function (CDF) or k-nearest-neighbors density estimation.

Related work

- (Rabanser, Günnemann, and Lipton 2019) explored shift detection through **statistical two-sample testing** with an empirical study on image datasets combining dimensionality reduction and two-sample testing for detecting distribution shifts in real-world ML systems.
- (Souza et al. 2020) addressed the limited availability of real-world data and **lack of benchmarks for adaptive classifiers** and drift detectors.

Conclusion

- Experimental results showed that using data drift detection to retrain the model enhanced the classifier's performance.
- While the detection methods did not immediately signal specific concept drifts, they effectively identified data drifts and prompted necessary classifier resets.
- Best results for the datasets were achieved with the smallest batch sizes analyzed.
- For future works, synthetic data and synthetic concept drifts will be introduced to show the effectiveness of monitoring data drift in concept drift scenarios.

Acknowledgments

This research was funded by grants CNPq proc. 420623/2023-0 and #2023/00779-0, São Paulo Research Foundation (FAPESP). It is also part of the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, FAPESP proc. 14/50937-1, and FAPESP proc. 15/24485-9.

Thank You!

https://helfs.me/assets/pdf/sbbd2024.pdf

References I

- Bland, J Martin and Douglas G Altman (1995). "Multiple significance tests: the Bonferroni method". In: *Bmj* 310.6973, p. 170.
- Dasu, Tamraparni et al. (2006). "An information-theoretic approach to detecting changes in multi-dimensional data streams". In: *Symposium on the Interface of Statistics, Computing Science, and Applications (Interface).*
- Ditzler, Gregory and Robi Polikar (2011). "Hellinger distance based drift detection for nonstationary environments". In: *2011 IEEE symposium on computational intelligence in dynamic and uncertain environments (CIDUE)*, pp. 41–48.

References II

- Lu, Jie et al. (2018). "Learning under concept drift: A review". In: *IEEE Transactions on Knowledge and Data Engineering* 31.12, pp. 2346–2363.
- Pérez-Cruz, Fernando (2008). "Kullback-Leibler divergence estimation of continuous distributions". In: *2008 IEEE international symposium on information theory*, pp. 1666–1670.
- Rabanser, Stephan, Stephan Günnemann, and Zachary Lipton (2019). "Failing loudly: An empirical study of methods for detecting dataset shift". In: *Advances in Neural Information Processing Systems* 32.

References III

Souza, V. M. A. et al. (2020). "Challenges in Benchmarking Stream Learning Algorithms with Real-world Data". In: *Data Mining and Knowledge Discovery* 34, pp. 1805–1858. DOI: 10.1007/s10618-020-00698-5.