Adaptive Aggregated Drift Detector

Beverly Abadines Quon [1] **Jean-Luc Gaudiot** [1]

Abstract

There needs to be an adaptive approach that combines both performance and distribution based concept drift detectors in order to harness the benefits of unlabeled data and the ability to detect varying types of drifts. This paper proposes Adaptive Aggregated Drift Detector (A2D2), which consists of a suite of performance and data distribution based detectors that can adaptively select detectors based on rankings of least cost. The notable contribution is that it enables an ecosytem to not only adaptively combat drift, but to also expand the information learned across a suite of detectors.

1. Introduction

Many machine learning strategies have a model-centric view which prioritizes fitting models onto a static dataset where performance (e.g. accuracy, F1 score) is often the main objective. Such models fail to consider the following: 1) testing data can be unstationary and drift from away from the context and concept originally captured under the training data and 2) performance comes with cost (Sculley et al., 2015). The former can be generalized as covariate drift (i.e. feature distributions diverge), label drift (i.e. class distributions diverge) and concept drift (i.e. relationship between feature and class diverges) (Lu et al., 2018). There is a collection of detectors that range from being performance based to data based distribution. Performance-based techniques keep track of a model's performance metrics, including recall, precision, F-measure, and accuracy. The cost is that they require labeled data, which can be expensive, making the expectation of abundant labeled data impractical. Data distribution-based techniques monitor changes in the location, density, and scope of the data itself rather than classifier performance parameters. These methods have the benefit of processing both labeled and unlabeled data, but they are restricted in the types of drift they can identify (Lu et al., 2019). Arguably, there needs to be an adaptive approach that combines both types in order to detect drift, specifically concept drift. Lastly, there needs to be a metric that considers the computational debt from retraining to fully mitigate its repercussions.

This paper proposes **Adaptive Aggregated Drift Detector (A2D2)**. It takes in a suite of detectors consisting of a mixture of performance based detectors and data distribution based (e.g. divergence) detectors. Its goal is to adaptively select the detector that optimizes N_{update} and $Gain_{\text{perf}}$ to achieve minimal costs. It applies exploitation and exploration strategies through the use of its Adaptive and Aggregative Phases.

Section 2 describes the utilization of a metric accounting for costs in terms of performance. Section 3 describes A2D2 and its metrics for evaluation under Section 4. Potential contributions under Section 5.

2. Preliminary Work

Quon & Gaudiot (2023) proposed **Performance Gained Update Cost Ratio (PGUCR)** Eq.(1) in order to relate a model's gain in performance with its cost of retraining updates in response to the detection of drift. PGUCR is normalized with values from 0 (ineffective) to 1 (effective). $F1_{new}$ represents the F1 score of a base classifier equipped with a detector. $F1_{ablation}$ is the score without any detectors and thereby without any updates to the model. N_{update} is the number of times a batch was triggered to retrain within a set number of batches. $Cost_{\text{update}}$ is an adjustable parameter representing the importance of updates in comparison to performance.

$$\text{PGUCR} = \frac{1}{2}\left(1 + \frac{F1_{\text{new}} - F1_{\text{ablation}}}{F1_{\text{ablation}}}\right) \div (1 + N_{\text{update}} \times Cost_{\text{update}}) \quad (1)$$

We manipulate the PGUCR to be in terms of $Cost_{update}$ and describe its upper and lower bounds. For clarity performance gain is:

$$Gain_{\text{perf}} = \frac{F1_{\text{new}} - F1_{\text{ablation}}}{F1_{\text{ablation}}} \quad (2)$$

[1]Electrical Engineering and Computer Science Department, University of California Irvine, Irvine CA 92697, USA. Correspondence to: Beverly Abadines Quon <babadine@uci.edu>, Jean-Luc Gaudiot <gaudiot@uci.edu>.

The lower bound of $Cost_{\text{update}}$ when PGUCR equals 1 simplifies to:

$$Cost_{\text{update}} = \frac{1}{2N_{\text{update}}}(Gain_{\text{perf}} - \frac{1}{2}) \tag{3}$$

The upper bound when PGUCR equals 0 signifies that regardless of any updates, the base classifier $Gain_{\text{perf}}$ is -1.

Our work applies the lower bound Eq.(3) which attempts to achieve max efficiency by minimizing costs improvements and PGUCR as metrics.

3. Methodology

3.1. Adaptive Phase

The Adaptive Phase dynamically selects the detector that has the best fit of achieving the highest PGUCR on top of generating new ensembles of detectors to add to the suite. It consists of the Terminal FCBF, Prequential Training and Testing, and Select & Explore units.

3.1.1. Terminal Fast Correlated Based Filtering (FCBF)

Fast Correlated Based Filtering (FCBF) (Yu & Liu, 2003; Nguyen et al., 2012) is a multivariate feature selection method that takes into account the dependencies between features and the class relevance. It uses information gain via entropy Eq.(4) to generate values under **Symmetrical Uncertainty (SU)** (Iserles, 1989). SU Eq.(6) calculates the dependencies between random variables X and Y such that it measures the effect of having knowledge on one has on the information learned from the other. SU is normalized between 0 (i.e. complete independence) to 1 (i.e. mutual dependence). $IG(X|Y)$ Eq.(5) denotes the information gain of X given Y.

$$H(X) = -\sum_{x} P(x_i)\log_2(P(x_i)) \tag{4}$$

$$\text{IG}(X|Y) = H(X) - H(X|Y) \tag{5}$$

$$SU(X,Y) = \frac{2 \times IG(X|Y)}{H(X) + H(Y)} \tag{6}$$

FCBF sorts features from highest to lowest SU ordering them from most to least relevance to the class. Yu & Lie (2003) iteratively removed redundant features by using the most predominant feature to heuristically compare and filter against lower valued features. Our method does not remove redundancies (hence the name Terminal FCBF) and keeps the SU values per features as a matrix. The generated SU matrix is then used as a blueprint for the data set in hand.

3.1.2. Prequential Training and Testing

The role of the base classifier is to predict the class, y based on the incoming features, X. Data is processed prequentially as windows, W with instances first used for testing followed by training. Every W is split into 10 batches, b. A selected detector processes through W and indicates if it suspects drift under any of b, serving as a list of triggers that the base classifier must retrain under. Each W is processed at least twice under the Adaptive Phase. Once for predicting under the triggered retrainings and another under ablation without any triggers. The F1 scores and triggers are sent to the Aggregative Phase.

3.1.3. Select & Explore units

Select takes the SU matrix and the Aggregated Embeddings (AgE) as inputs in order to predict which existing detector is the best for current W. The Explore unit takes in the selected detector, DD and references the Collaborative Filtering Recommender System (CFRS) in order to identify which detector has the least similarity with (if any). If there exists a pair, the union of triggers between DD and the complementary detector, DD' are used. If an ensemble is created, then the classifier must run for a third time and subsequently compares if $\text{PGUCR}_{DD} \leq \text{PGUCR}_{DD'}$ noting that DD' is worth adding to the suite.

3.2. Aggregative Phase

The Aggregative Phase develops a collective knowledge of each detector with respect to the W processed. If the Adaptive Phase is considered as online testing, then the Aggregative Phase would be considered the offline training. Hence W, which was the current data from the Adaptive Phase is viewed as the reference data under the Aggregative Phase. Under this phase, all the detectors take turns detecting drift and measuring their performance under the Detector Test Suite. The Detector Test Suite works similarly as the Adaptive Phase, where classifier with detector is compared with the ablation test. The Detector Test Suite outputs the F1 score and triggers to the Cost Based Ordinality (CBO) and only the triggers to CFRS.

3.2.1. Collaborative Filtering Recommender System

CFRS takes the key value pairs of detectors and triggers and represents them into a matrix, where rows indicate detectors and columns indicate whether a drift was detected at b.It applies the Jaccard similarity coefficient (Ivchenko & Honov, 1998) to calculate the pairwise similarities amongst detectors and outputs their similarity scores. Unorthodoxically, CFRS is used for finding the least similar detector for the DD under the Adaptive Phase.

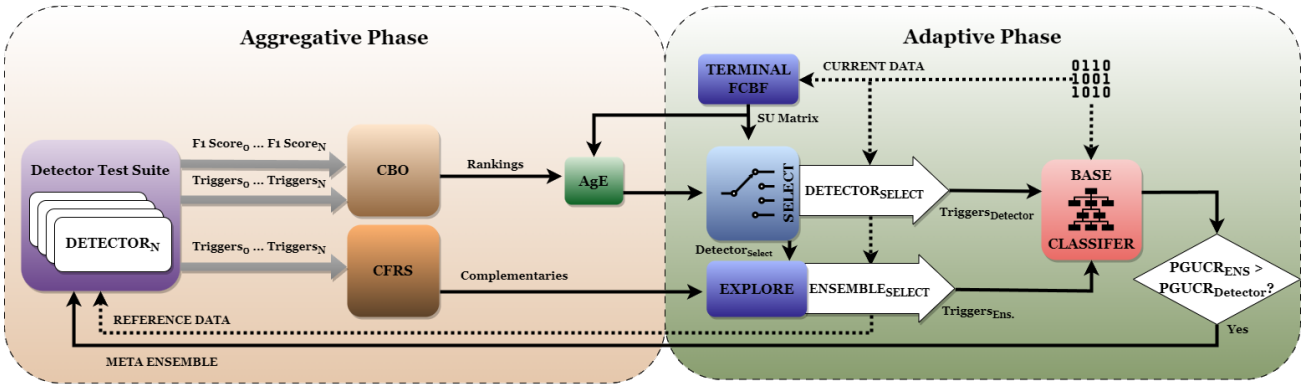

Figure 1. A2D2 architecture has Adaptive and Aggregative Phases. The Adaptive Phase applies exploitation and exploration strategies in order to select the most cost effective detector out of the suite of detectors. Simultaneously it creates an ensemble of detectors composed of the selected detector and its most complimentary detector (i.e. detector with the least similarity in terms of batches triggered). The Aggregative Phase updates its Aggregated Embeddings (AgE) via the SU matrix and the updated rankings from the Cost Based Ordinality (CBO). To generate complimentary ensembles, batches triggered per detector are inputted to the Collaborative Filtering Recommender System (CFRS),which scores detectors of their similarities. If the newly created ensemble detector is more cost effective than the selected detector, then the ensemble grouping is added to the suite of detectors.

3.2.2. Cost Based Ordinality

The CBO ranks the detectors according to Eq.(3). Heuristically, the detectors are ranked from largest to smallest cost, but an alternative method not implemented is to have the detectors fall under rankings based on where their $Cost_{\text{update}}$ falls under the intervals of $1/N_{\text{Detectors}}$, where $N_{\text{Detectors}}$ is the current number of detectors in the suite. The key value pairs of detector to ranking are processed to AgE.

3.2.3. Aggregated Embeddings

AgE saves the SU matrix from the Adaptive Phase and connects it with the rankings after the Detector Test Suite and CBO are complete. AgE appends the table of SU with the detector-rankings, where SU can be interpreted as the features that predict the classification defined by the detector-rankings. In other words, AgE is used to predict which detectors would have the smallest $Cost_{\text{update}}$ under a specific SU.

4. Evaluation

This work will use the MOA framework (Bifet et al., 2010; 2013) to evaluate A2D2 on 3 artificial datasets and 2 real-world datasets that were injected with concept drift. Our method will include 6 divergence tests capable of detecting changes in distribution specifically, Cramer Von Mises test (CMV), Energy Distance test (EDT), Kolmogorov-Smirnov test (KS), Mann-Whitney U-rank test (MWT), T test, and Wasserstein Distance test (WD). It will also include 6 performance based detectors DDM, EDDM, KSWIN, PH, ADWIN, HDDM_A, and HDDM_W as part of the Detector Test Suite.

4.1. Artifical Dataset Configuration

MOA can inject concept drift by connecting data streams as a weighted combination of distributions whose likelihood of an instance originating from the new concept is defined by a sigmoid function. Each dataset will contain 10K instances with widths ranging from 0.5K to 4K instances. The drift's midpoints will fall in between 1.5 and 7.5 kilometers. By altering the instantiation of streams using a random seed, 10 tests will be created from each dataset.

4.2. Artificial Datasets

Agrawal (Agrawal et al., 1993) describes 6 numerical and 3 categorical features mapped to 10 different pre-defined loan functions

LED (Bifet et al., 2009; Schlimmer & Granger, 1986; Breiman et al., 1984) comprises of 24 binary attributes, but only 7 are relevant for predicting the next digit on a seven-segment LED display (i.e. 10 classes).

Sea (Bifet et al., 2009; Schlimmer & Granger, 1986; Street & Kim, 2001) uses 3 attributes, but 1 is irrelevant. All attributes have values between 0 and 10, and comparing the sum of the relevant attributes with a threshold parameter determines which of the 4 classes it is mapped to.

4.3. Real World Datasets

The following datasets are nomalized by MOA, so that the numerical values are between 0 and 1.

Electricity (Harries, 1999) contains 8 attributes to predict whether the Australian New South Wales Electricity Market from May 1996 to December 1998 rises or falls (i.e. 2 classes). The dataset contains 45,312 instances.

Poker is a modified version of (Cattral et al., 2001) without duplicates and is sorted by rank and suit. Each instance is a hand of 5 cards drawn from a 52 card deck. It is made up of 10 attributes (rank and suit of cards in hand) to predict 10 poker hands or classes. The dataset has 1,000,000 instances.

5. Conclusion

The potential advantages of A2D2 will be its ability to 1) embed W as training data to predict rankings of detectors and adaptively select the one with least cost and 2) develop a collective understanding of detectors that continues to grow. The notable contribution may be 2) as it enables an ecosytem to not only adaptively combat drift, but to also expand the information learned across a suite of detectors. Although A2D2 is left to be implemented we have collected preliminary empirical work related to the Detector Test Suite, which is included in the appendix.

References

Agrawal, R., Imielinski, T., and Swami, A. N. Database mining: A performance perspective. *IEEE Trans. Knowl. Data Eng.*, 5(6):914–925, 1993. doi: 10.1109/69.250074. URL https://doi.org/10.1109/69.250074.

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., and Gavaldà, R. New ensemble methods for evolving data streams. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009. doi: 10.1145/1557019.1557041.

Bifet, A., Holmes, G., Kirkby, R., and Pfahringer, B. MOA: massive online analysis. *J. Mach. Learn. Res.*, 11:1601–1604, 2010. doi: 10.5555/1756006.1859903. URL https://dl.acm.org/doi/10.5555/1756006.1859903.

Bifet, A., Read, J., Pfahringer, B., Holmes, G., and Žliobaitė, I. Cd-moa: Change detection framework for massive online analysis. In *Advances in Intelligent Data Analysis XII: 12th International Symposium, IDA 2013, London, UK, October 17-19, 2013. Proceedings 12*, pp. 92–103. Springer, 2013.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. *Classification and Regression Trees*. Wadsworth, 1984. ISBN 0-534-98053-8.

Cattral, R., Oppacher, F., and Deugo, D. Evolutionary data mining with automatic rule generalization. 2001.

Harries, M. B. Splice-2 comparative evaluation: Electricity pricing. 1999.

Iserles, A. Numerical recipes in c—the art of scientific computing, by w. h. press, b. p. flannery, s. a. teukolsky and w. t. vetterling. pp 735. £27·50. 1988. isbn 0-521-35465-x (cambridge university press). *The Mathematical Gazette*, 73(464):167–170, 1989. doi: 10.2307/3619708.

Ivchenko, G. and Honov, S. On the jaccard similarity test. *Journal of Mathematical Sciences*, 88:789–794, 1998.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. Learning under concept drift: A review. *IEEE Trans. Knowl. Data Eng.*, 31(12):2346–2363, 2019. doi: 10.1109/TKDE.2018.2876857. URL https://doi.org/10.1109/TKDE.2018.2876857.

Nguyen, H., Woon, Y., Ng, W. K., and Wan, L. Heterogeneous ensemble for feature drifts in data streams. In *Advances in Knowledge Discovery and Data Mining - 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29 - June 1, 2012, Proceedings, Part II*, volume 7302 of *Lecture Notes in Computer Science*, pp. 1–12. Springer, 2012. doi: 10.1007/978-3-642-30220-6_1. URL https://doi.org/10.1007/978-3-642-30220-6_1.

Quon, B. A. and Gaudiot, J.-L. COLLABORATIVE CONCEPT DRIFT DETECTION, 2023. URL https://openreview.net/forum?id=STpRX-XCO6t.

Schlimmer, J. C. and Granger, R. H. Incremental learning from noisy data. *Mach. Learn.*, 1(3):317–354, 1986. doi: 10.1023/A:1022810614389. URL https://doi.org/10.1023/A:1022810614389.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., and Dennison, D. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28, 2015.

Street, W. N. and Kim, Y. A streaming ensemble algorithm (SEA) for large-scale classification. In Lee, D., Schkolnick, M., Provost, F. J., and Srikant, R. (eds.), *Proceedings of the seventh ACM SIGKDD international*

conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001, pp. 377–382. ACM, 2001. doi: 10.1145/502512.502568. URL https://doi.org/10.1145/502512.502568.

Yu, L. and Liu, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 856–863, 2003.

Table 1. F1 Scores of HFT on synthetic datasets that were retrained on batches identified by divergence tests. ABL refers to ablation test.

TEST	AGRAWAL	LED	SEA
CVM	1 ± 0.00	0.66 ± 0.04	0.84 ± 0.01
ED	1 ± 0.00	0.64 ± 0.05	0.84 ± 0.01
FCBF	1 ± 0.00	0.65 ± 0.05	0.84 ± 0.01
KS	1 ± 0.00	0.65 ± 0.04	0.84 ± 0.02
MW	1 ± 0.00	0.66 ± 0.04	0.83 ± 0.02
T	1 ± 0.00	0.66 ± 0.04	0.83 ± 0.02
WD	1 ± 0.00	0.66 ± 0.04	0.83 ± 0.02
ABL	1 ± 0.00	0.58 ± 0.10	0.83 ± 0.02

Table 2. F1 Scores of HFT on real datasets that were retrained on batches identified by divergence tests. ABL refers to ablation test.

TEST	AIRLINES	ELECTRICITY	POKER
CVM	0.56	0.70	0.21
EDT	0.56	0.70	0.21
FCBF	0.47	0.64	0.16
KS	0.56	0.70	0.21
MWT	0.56	0.70	0.21
T	0.56	0.70	0.21
WDT	0.56	0.70	0.21
ABL	0.56	0.55	0.14

Table 3. Average number of triggers for updates on synthetic datasets. Ablation excluded as the count will always be zero.

TEST	AGRAWAL	LED	SEA
CVM	4.3 ± 0.9	8.1 ± 0.9	2.0 ± 0.7
EDT	9.0 ± 0.0	2.0 ± 0.7	4.0 ± 0.9
FCBF	3.1 ± 1.4	3.6 ± 1.3	1.6 ± 0.5
KS	3.5 ± 1.8	3.6 ± 1.7	1.8 ± 0.8
MWT	5.4 ± 0.5	7.7 ± 2.1	1.8 ± 0.4
T	6.0 ± 0.7	8.3 ± 0.7	1.8 ± 0.4
WDT	3.0 ± 0.7	4.7 ± 0.9	0 ± 0.00

Table 4. Number of triggers for updates on real datasets. Ablation excluded as the count will always be zero.

TEST	AIRLINES	ELECTRICITY	POKER
CVM	9	9	9
EDT	9	9	9
FCBF	1	3	3
KS	9	9	9
MWT	9	9	9
T	9	9	9
WDT	9	9	9

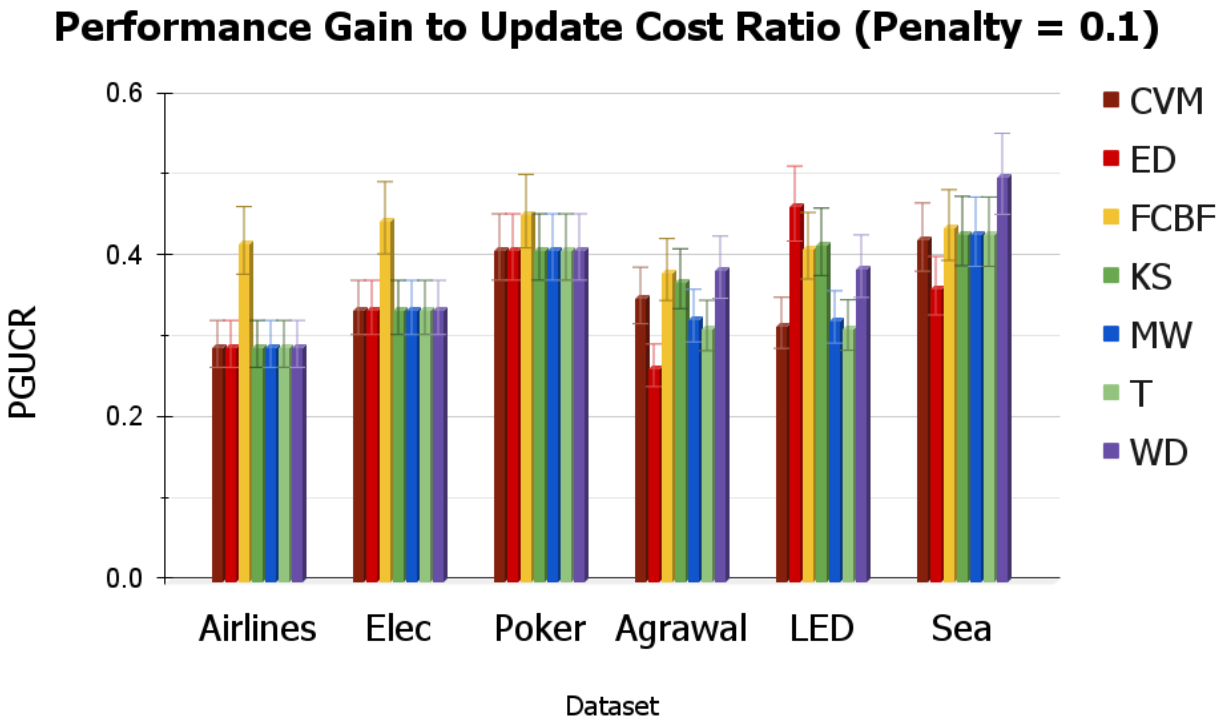

Figure 2. Accuracy Gain to Update Cost Ratio when the penalty for updates is 0.1. Values closer to 1 are more cost effective

Table 5. AGUCR (penalty =0.1) for real and synthetic datasets with rankings

TEST	MEAN AGUCR	MEAN RANK
CVM	0.36 ± 0.05	5.33
ED	0.36 ± 0.15	5.00
FCBF	0.41 ± 0.03	2.33
KS	0.41 ± 0.03	2.67
MW	0.36 ± 0.06	4.83
T	0.35 ± 0.07	5.83
WD	0.42 ± 0.7	2.00
χ_F^2=9.90	F_F=2.44	CRITICAL VALUE AT α=0.1 IS 2.33

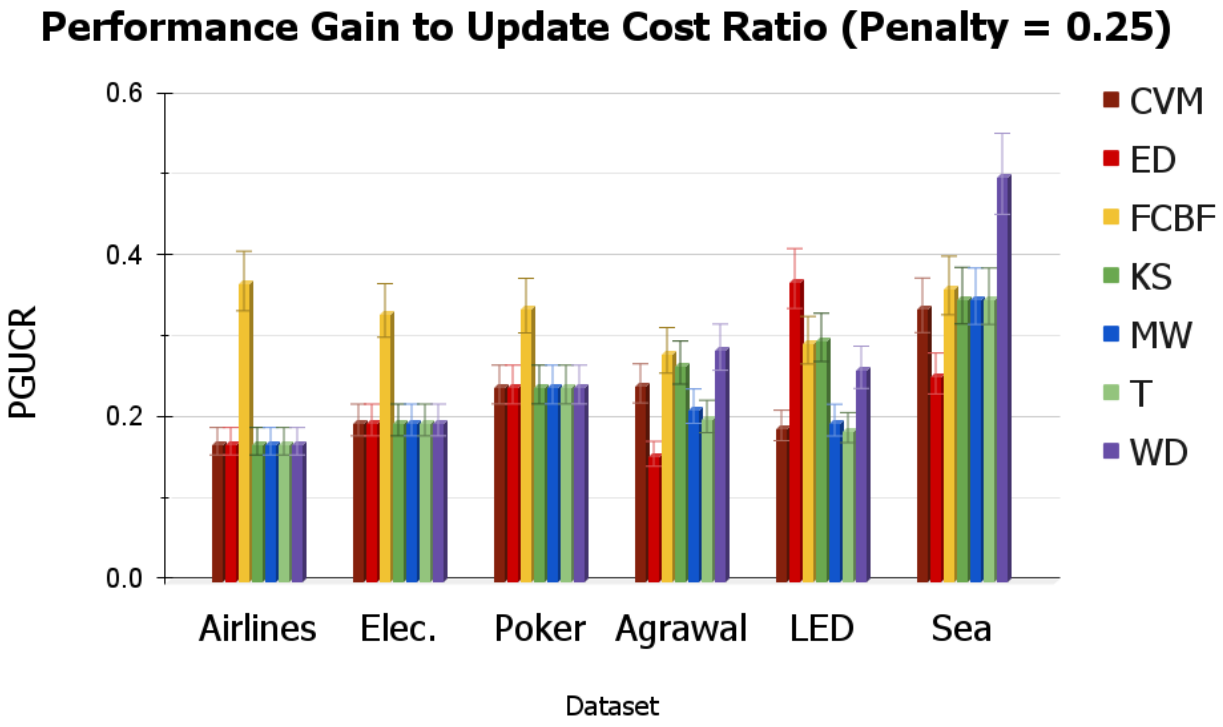

Figure 3. Performance Gain to Update Cost Ratio when the penalty for updates is 0.25. Values closer to 1 are more cost effective

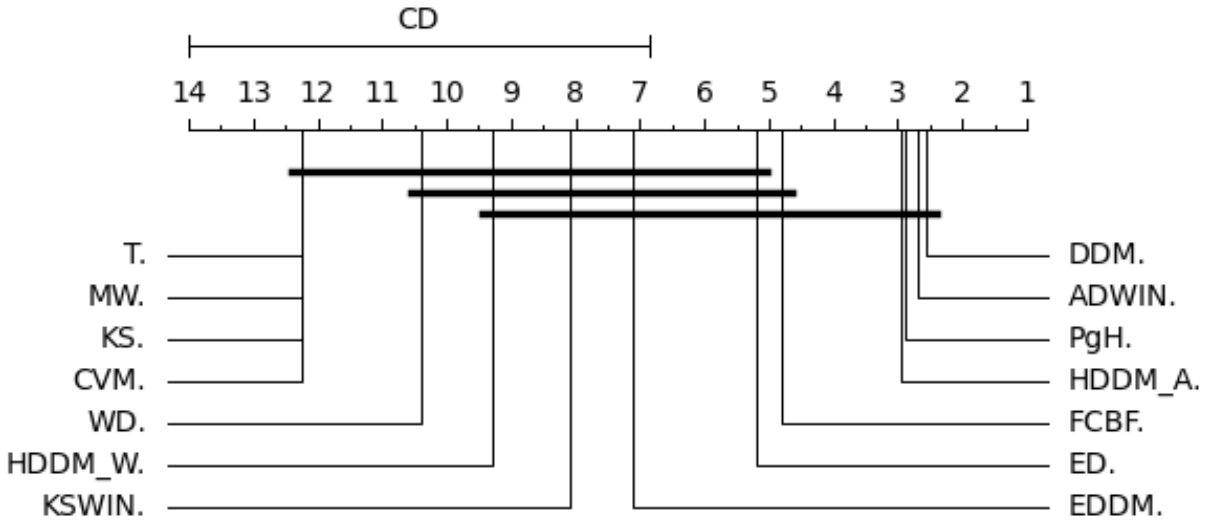

Figure 4. Post-hoc Nemenyi test at confidence level of 99% of the AGUCR from RandTree100 dataset. Critical distance is 7.144. With respect to FCBF there is a significant difference between it and T, MW, KS, and CVM.