DM-Lab: Heterogeneous Ensemble for Feature Drifts in Data Streams

Jan Kohlhase, Sebastian Döhler, Noah Wöhler

Gottfried Wilhelm Leibniz Universität Hannover

{*jan.kohlhase, sebastian.doehler, woehler*}*@stud.uni-hannover.de*

February 28, 2020

Introduction I

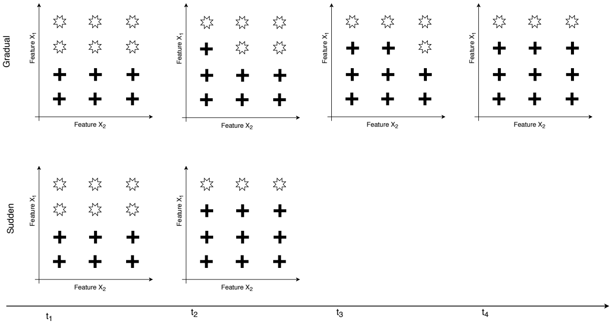
- Modern real-life applications produce massive amounts of data e.g. online stores, stock markets
- Needs to be processed to gain information from it
- Underlying concept of the data changes over time

Data stream classification requirements

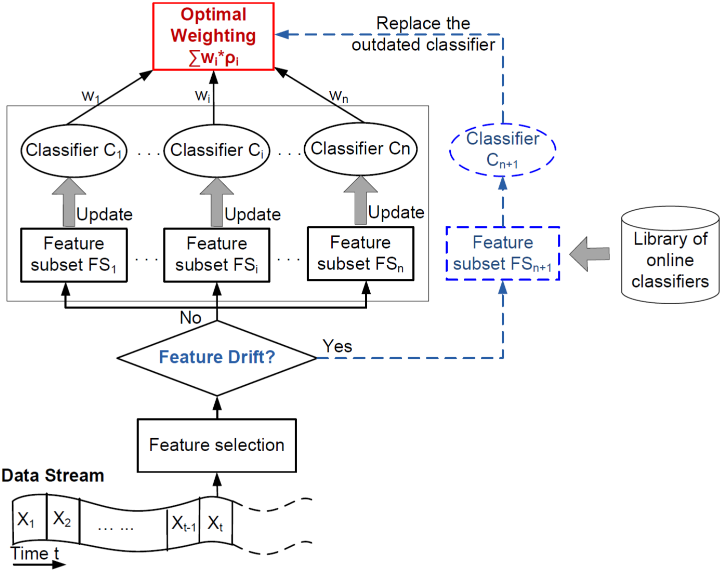
- Real-time
- Efficient
- Deal with high-dimensional continuously arriving data

Introduction II

- The patterns discovered in data can change over time.
- Commonly referred to as concept drift
- Two different kinds of changes:
  - Gradual drift
  - Sudden drift



Overview of the algorithm



Method

Three main building blocks

- Selection of relevant features and detection of concept drifts
- Choice of multiple classifiers to create a heterogeneous ensemble
- Optimal weighting and retention of the best classifiers used in the ensemble

Method: Feature Selection

- Apply any feature selector $\rightarrow$ Subspace of critical features
- Compare feature subspaces at consecutive points in time
  - Subspaces differ $\rightarrow$ Feature drift $\rightarrow$ Train new classifier (best performing type)
  - Subspace remained constant $\rightarrow$ No sudden drift (likely gradual) $\rightarrow$ Partially fit online classifiers on new chunk

Feature selector in HEFT: *Fast Correlation-Based Filter (FCBF)*

- Simple, fast, effective
- Selects features that correlate strongly with the class label
- Leaves out redundant features that correlate strongly with each other

Method: Ensemble

- Combine several weak learners' predictions to form a strong classifier
- Easy, efficient method to improve classification results
- (few) classifiers of different types → **heterogeneous**

Initial ensemble: one of each base learner (Online Naive Bayes, CVFDT)

- Ensemble not full → train and add new classifier of best-performing type
- Ensemble full → remove outdated, i.e. worst-performing classifier first

Method: Optimal Weighting

- Each classifier in the ensemble has an associated weight
- Weights are updated on each classification iteration according to
  - Accuracy of the classifier
  - Accumulated error from its creation time
- Apply online bagging for updating individual classifiers $\rightarrow$ Reduce variance of the ensemble
- Final prediction is a weighted sum of all classifiers' predictions

Comparison with the state of the art

Positive:

1. Faster training time due to feature selection
2. Adapts to both types of drifts by replacing outdated classifiers for sudden drifts and partially fitting the learners for gradual drifts.
3. Diversity in the classification, because of the heterogeneity of the ensemble

Negative:

1. Longer training time when the ensemble is big
2. The method converges to a homogeneous ensemble when training long enough

Implementation

Implementation with the machine learning framework *scikit-multiflow*

- Based on the *Average Weighted Ensemble* (AWE)
- Feature selection with *Fast Correlation-Based Filter* (FCBF) and *Conditional Mutual Information* (CMIM)
- Ensemble learning with feature drift detection
- Ensemble classification with aggregated error

Experimental aim

Test the influence of different parts of the proposed algorithm regarding runtime and classification accuracy

- Base learners
- Feature selection method
- Ensemble construction

Experiments: Datasets

Datasets used for the evaluation of the HEFT method. All experiments were conducted on 25,000 samples and with a batch size of 1,000 samples.

Synthetic:

- LED *24 features, 10 classes*: seven-segment display classification
- SEA *3 features, 10 classes*: sum of 2 relevant features compared to a threshold

Real World:

- KDD *41 features, 23 classes*: network connection type classification
- WEATHER *8 features, 2 classes*: rain prediction

Experimental setup: Different base learners

Different base learners tested with HEFT

- Online Naive Bayes (NB)
- Extremly Fast Decision Tree (HATT)
- Multilayer Perceptron (MLP)
- Proposed setup: NB & HATT

Comparison with other ensemble methods provided by the scikit-multiflow framework

- Average Weighted Ensemble (AWE)
- Additive Expert Ensemble (AEE)
- Baseline: AWE

Experimental results: Different base learners

Experimental results on synthetic and real-world datasets

	HEFT		AWE(NB)		AEE(NB)	
	Acc	Time[s]	Acc	Time[s]	Acc	Time[s]
LED	**0.733**[1]	<u>99.2</u>[1]	0.729	463.5	**0.733**	579.3
SEA	0.825[2]	44.2[2]	0.830	<u>31.3</u>	**0.881**	42.3
KDD	0.843[3]	636.2[3]	0.742	<u>152.4</u>	**0.995**	596.9
weather	0.737[4]	173.3[4]	**0.738**	<u>20.1</u>	0.694	27.2

[1] Base learners: NB, MLP; non-random choice; CMIM feature selection
[2] Base learners: NB, MLP; non-random choice; CMIM feature selection
[3] Base learners: NB, HATT; non-random choice; FCBF feature selection
[4] Base learners: NB, HATT, MLP; non-random choice; CMIM feature selection

Experimental setup: Different ensemble creation methods

Different ensemble construction methods tested with HEFT

- Best performing base learner (best)
- Randomly chosen base learner (random)
- Proposed setup: best

Experimental results: Different ensemble creation methods

Experimental results for different ensemble construction methods

	standard		random	
	Acc	Time[s]	Acc	Time[s]
LED	0.624	1578.7	**0.680**	1731.0
SEA	**0.714**	152.1	0.662	205.0
KDD	**0.736**	343.2	**0.736**	404.3
weather	0.696	71.8	**0.705**	87.4

Experimental setup: Different feature selectors

Different feature selection methods tested with HEFT

- Fast Correlation Based Filter (FCBF)
- Conditional Mutual Information (CMIM)
- Proposed setup: FCBF

Experimental results: Different feature selectors

Experimental results for different feature selection methods

	FCBF		CMIM	
	Acc	Time[s]	Acc	Time[s]
LED	0.583	<u>217.0</u>	**0.721**	3092.7
SEA	0.660	<u>152.9</u>	**0.716**	204.1
KDD	**0.843**	676.3	0.630	<u>71.2</u>
weather	0.687	<u>23.2</u>	**0.714**	136.1

Conclusion

- Mean accuracy of our implementation close to other state-of-the-art stream ensemble learning algorithms
- Runtime is significantly worse
- Reproduction of the authors' gains was not possible on every dataset
- Random method for ensemble creation has a positive impact on real-world dataset (accuracy), negative impact on artificial datasets
- However, it drastically reduces the chance of converging to a homogeneous ensemble
- CMIM performs well on numeric feature datasets (accuracy) but tends to select all features
- FCBF better on nominal feature datasets and much faster to compute, because it selects fewer features

GitHub

- Page: https://kohlhasej.github.io/heftstream
- Code: https://github.com/KohlhaseJ/heftstream

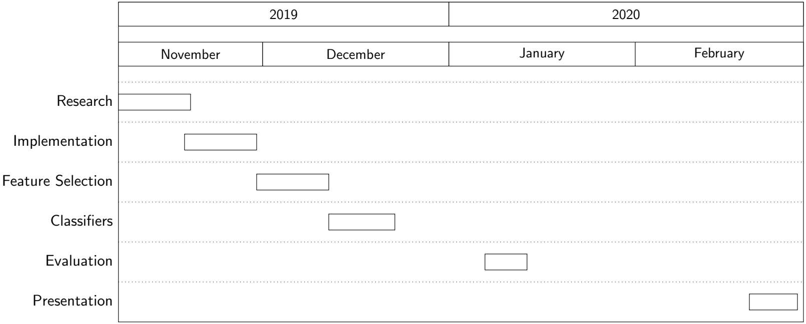
References

Hai-Long Nguyen, Yew-Kwong Woon, Wee-Keong Ng, Li Wan (2012)
Heterogeneous ensemble for feature drifts in data streams
Pacific-Asia conference on knowledge discovery and data mining, 1 – 12.

[Jacob Montiel, 2018] Jacob Montiel, Jesse Read, Albert Bifet, Talel Abdessalem (2018)
Scikit-Multiflow: A Multi-output Streaming Framework
Journal of Machine Learning Research, Vol 19, no 72, 1 – 5

[François Fleuret, 2004] François Fleuret (2004)
Fast binary feature selection with conditional mutual information
Journal of Machine Learning Research, Vol 5, 1531 – 1555

Timetable


2019
2020
November
December
January
February
Research
Implementation
Feature Selection
Classifiers
Evaluation
Presentation