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

- Modern real-life applications produce massive amounts of data
- Needs to be processed to gain information from it
- Underlying concept of the data changes over time

Data stream classification requirements

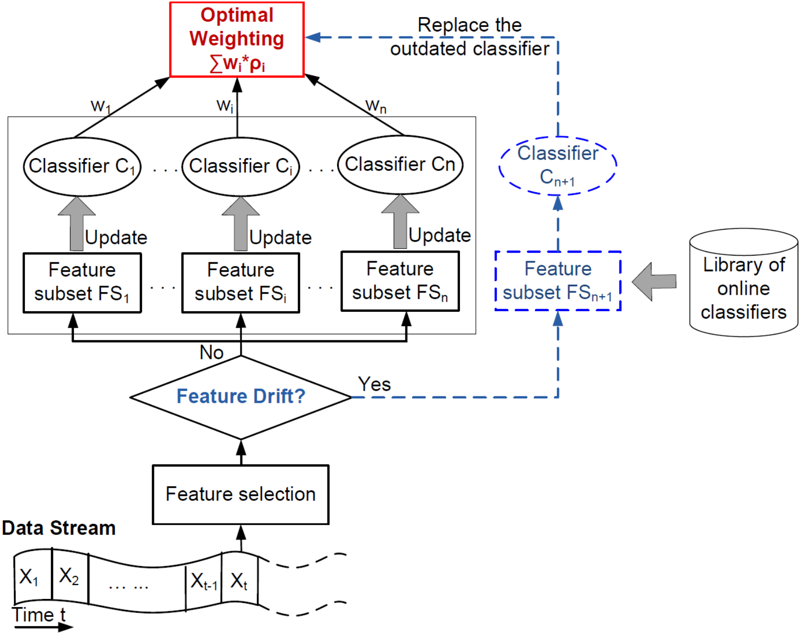
- real-time
- efficient
- deal with high-dimensional continuously arriving data

Introduction

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

Data stream classification needs to be real-time, efficient and be able to cope with high-dimensional, continuously arriving data. This problem occurs in many real-life applications like online stores or stock markets. A lot of modern technologies produce massive amounts of data that need to be processed to gain information from it. The discovered patterns from this data change over time. Two different kinds of changes exist that are commonly referred to as concept drifts. There are gradual drifts with small changes over time and sudden drifts that incur drastic changes. To deal with this, HEFT proposes a method to incorporate feature selection into ensemble learning to reduce computational complexity and adapt to the different types of concept drifts. For the feature selection, a

Overview of the algorithm



By comparing the feature subspaces at two consecutive points in time, the algorithm detects feature drifts. If the current feature subspace differs from the previous, then there is a change in the underlying distribution. It then applies an ensemble block which is an easy and efficient method to improve the classification results. A small heterogeneous ensemble is used instead of a homogeneous one. The optimal weighting method assigns each classifier a weight with respect to its classification error. According to this weight, the final prediction is a weighted sum of all the classifiers' predictions. If there was a feature drift, then a newly trained classifier of the best performing type is added to the ensemble. If no feature drift occurred, then all the classifiers in the ensemble are partially fitted on the current chunk of data. If the ensemble is full and a new classifier is added to the ensemble, the worst performing classifier is removed.

Feature selection

There are two different types of drifts that we distinguish. First, there is a *concept drift* which describes changes in the probability distribution of the classes at two consecutive time points. Second, there is a *feature drift* which is a change in the most discriminative feature subset at two consecutive time points.

- Selection of relevant features: Maximize the correlation with the class label and minimize the correlation with other features
- Dimensionality reduction: not all features in high-dimensional datasets are critical for training a classifier

Fast Correlation Based Filter (FCBF) selects features that correlate strongly with the class label, but have little correlation with other features. The correlation measure used is called Symmetrical Uncertainty.
Conditional Mutual Information (CMIM) is similar to FCBF. It uses a cost function that focuses on both correlation with the class label and independence from other features and does not require tuning of a threshold hyperparameter.

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

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

Experimental setup

Different base learners tested with HEFT

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

Experimental setup

Different feature selection methods tested with HEFT

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

Experimental setup

Different ensemble construction tested with HEFT

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

Experimental setup

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

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

Experimental setup

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*:

Real World:

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

Experimental results

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]	99.2[1]	0.729	463.5	**0.733**	579.3
SEA	0.825[2]	44.2[2]	0.830	31.3	**0.881**	42.3
KDD	0.843[3]	636.2[3]	0.742	152.4	**0.995**	596.9
weather	0.737[4]	173.3[4]	**0.738**	20.1	0.694	27.2

Experimental results

Experimental results for different ensemble construction

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

Experimental results

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

How does the proposed method improve the state of the art?

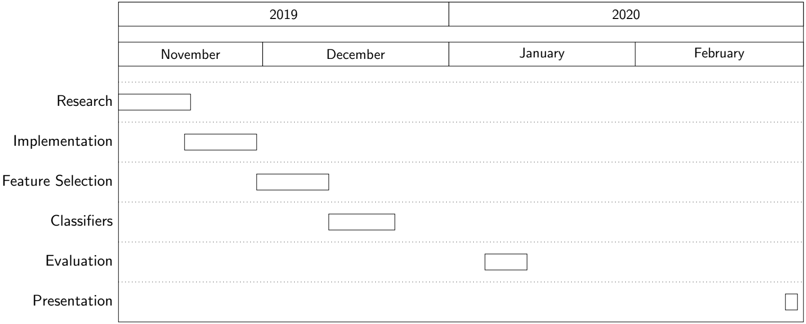
Positive:

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

Timetable



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

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