Algorithm Selection on Data Streams

Jan N. van Rijn [1] Geoffrey Holmes [2] Bernhard Pfahringer [2] Joaquin Vanschoren [3]

[1]Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands
[2]Department of Computer Science, University of Waikato, Hamilton, New Zealand
[3]Department of Mathematics and Computer Science, TU Eindhoven, Eindhoven, The Netherlands

Introduction

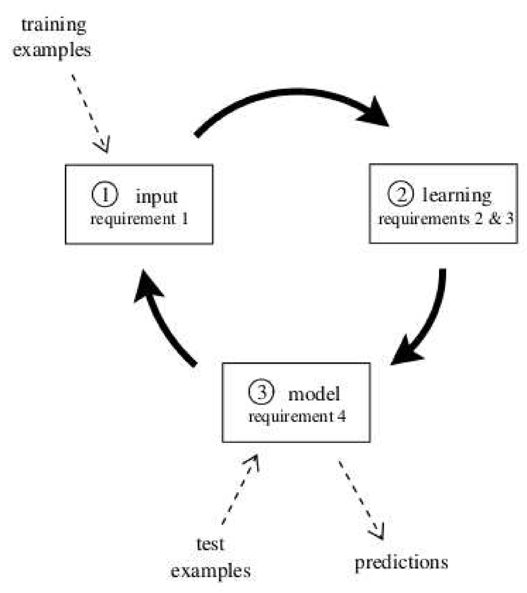
Modern society produces vast amounts of data coming, for instance, from sensor networks and various text sources on the internet. Various machine learning algorithms are able to capture general trends and make predictions for future observations with a reasonable success rate.
Using Meta-Learning, our goal is to establish knowledge about what type of algorithm works well on which kind of data.

Data Streams

Requirements: (as stated by [1])

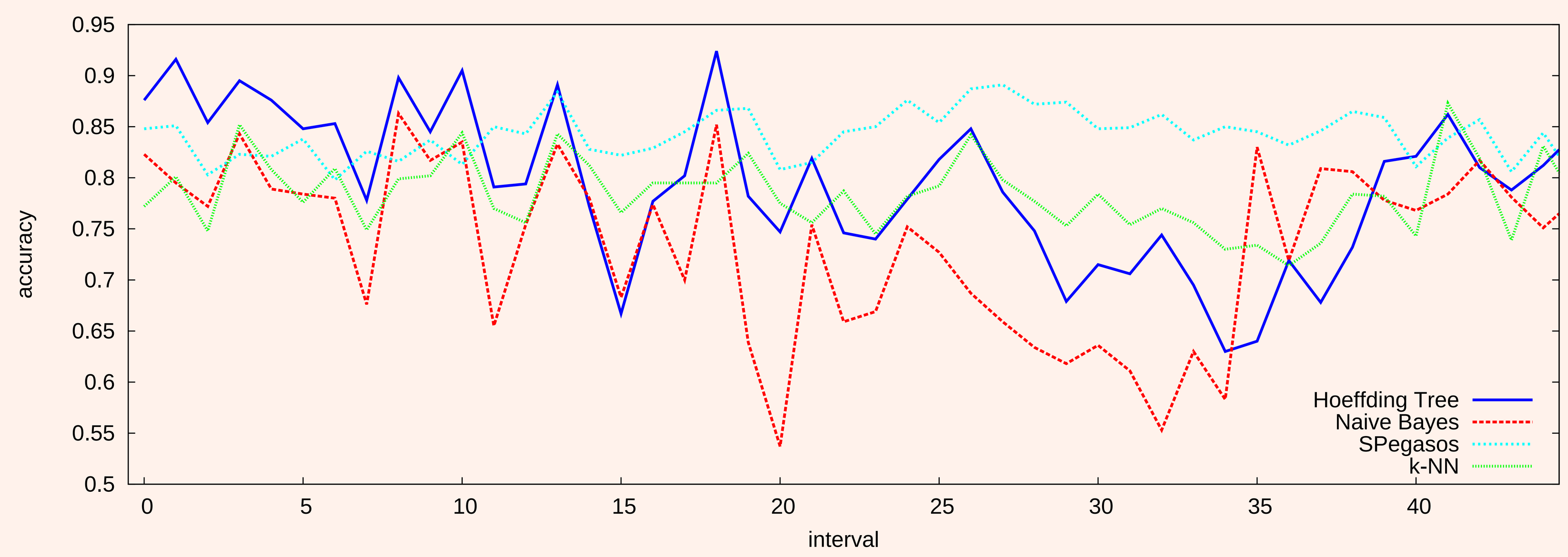
- Process an example at a time, and inspect it only once
- Use a limited amount of memory
- Work in a limited amount of time
- Be ready to predict at any point

Interleaved test-then-train: Each instance is first used to test the model, after which it can be used to train the model. E.g.: The stock market.



Meta-Learning

It is common for learning curves in Data Streams to cross multiple times, as illustrated by this plot, describing the *Electricity* data stream.



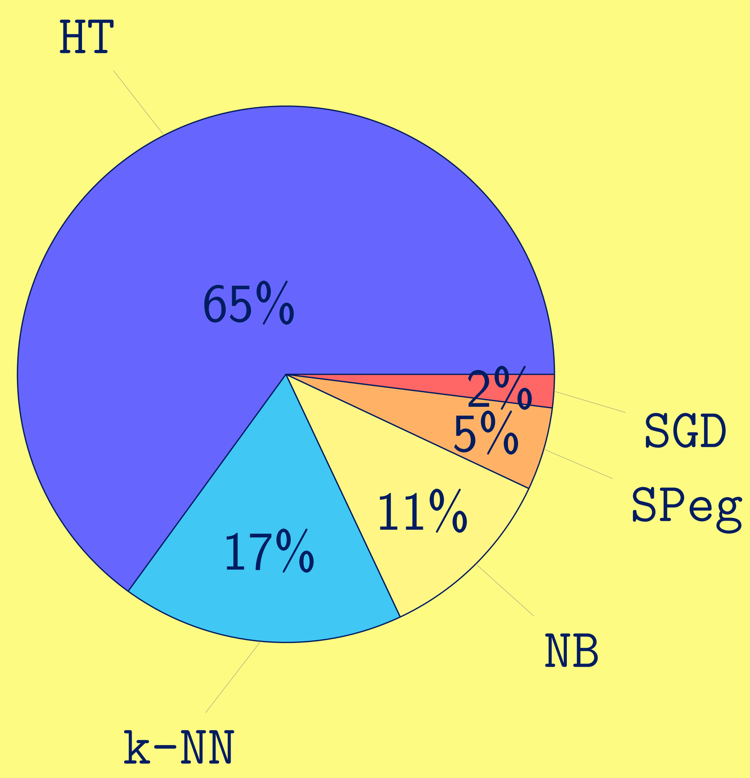
Idea: What if we could determine dynamically at any point in the stream which algorithm to use

Algorithm Selection

- Run a set of algorithms full factorial on a set of data streams.
- All algorithms as implemented in the MOA framework [1].
- We used real world data streams, data generators and semi generated data.
- Split data streams up in windows of n instances.
- Calculate a set of *meta-features* of each interval (SSIL).
- Introduced novel stream-based meta-features.
- Store evaluation scores of each algorithm on each interval.
- All results available on `http://www.openml.org/` [3].

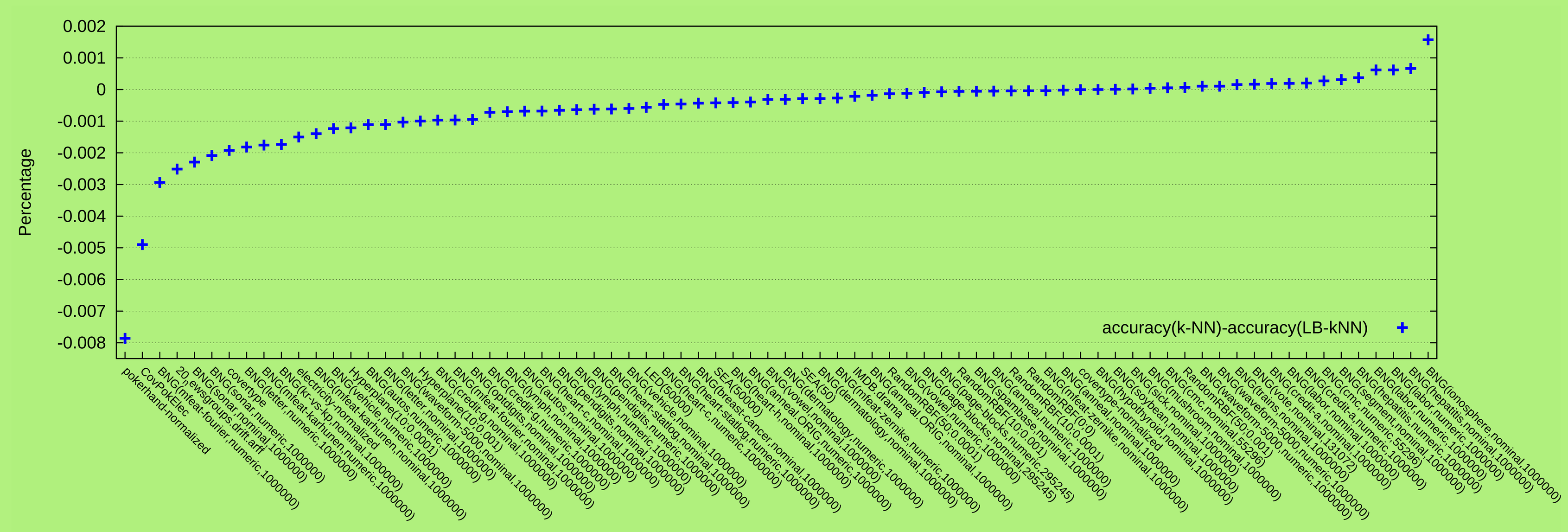
Discovery 1

Using basic meta-features, we can predict the best performing classifier from the following set of classifiers: {`Naive Bayes`, `k-NN`, `Hoeffding Tree`, `SPegasus`, `Stochastic Gradient Descent`}.
The pie chart shows the skewness of this dataset. `Hoeffding Trees` perform best on most intervals. Measured over all data streams, a meta-classifier was able to predict the best performing classifier in more than 80% of the cases. It improved the baseline on both *meta-level accuracy* and *base-level accuracy*. The results are competitive with state of the art meta-classifiers.



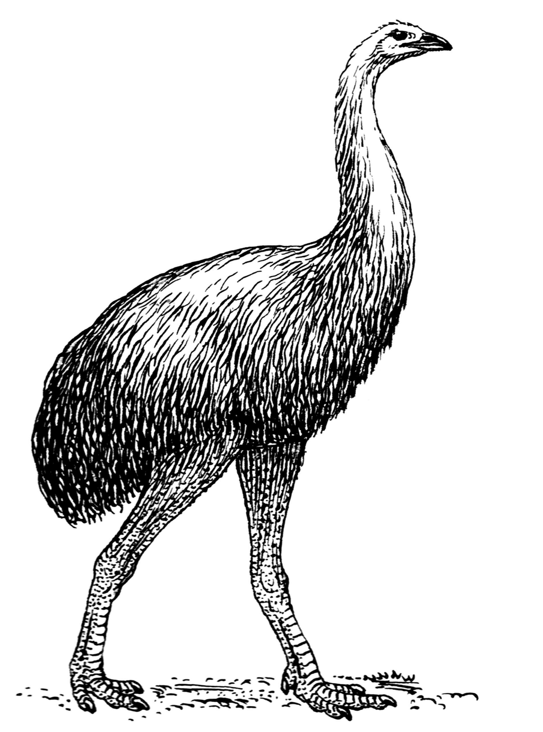
Discovery 2

`Leveraging Bagging` is a state of the art meta-learning technique, proven superior to other algorithms many times. It is in particular successful when used on `Hoeffding trees`.



This plot shows that applying `Leveraged Bagging` to `k-NN` actually decreases the performance! Breiman [4] reported on similar behaviour in the classical setting.

Discovery 3



`Naive Bayes` works well when the meta-feature measuring the number of changes detected in the stream is high. `Naive Bayes` generally needs only relatively few observations to achieve good accuracy compared to more sophisticated algorithms such as `Hoeffding Trees`. Assuming that a high number of changes detected by this landmarker indicates that the concept of the stream is indeed changing quickly, this could explain why a classifier like `Naive Bayes` outperforms more sophisticated learning algorithms that need more observations of the same concept to perform well.

References

[1] A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer. MOA: Massive Online Analysis. J. Mach. Learn. Res. 11, pages 1601–1604, 2010.
[2] J. N. van Rijn, G. Holmes, B. Pfahringer, and J. Vanschoren. Algorithm Selection on Data Streams. Discovery Science, 17th International Conference, pages 325–336, 2014.
[3] J. Vanschoren, J.N. van Rijn, B. Bischl and L. Torgo. OpenML: networked science in machine learning. ACM SIGKDD Explorations Newsletter 15, pages 49–60, 2014
[4] L. Breiman. Bagging Predictors. Machine learning 24(2), pages 123–140, 1996.

Technische Universiteit Eindhoven University of Technology