Data Stream Mining- Lecture 4

Classification of Data Streams

Chandresh Kumar Maurya, Assistant Professor

Eötvös Loránd University, Budapest, Hungary

October 15, 2019

Classification of data streams

Classification of static data- several algorithms exists such as NN, KNN, decision tress, SVM etc.
Classification of dynamic streams is non-trivial due to the following challenges:

1. Concept drifts
2. High data rate
3. Low memory footprints
4. Low processing time
5. Single (or 2-3 passes)
6. Algorithms needs to be robust
7. and so on.

Traditional algorithms needs to be adapted to account for the aforementioned issues.

Solution Techniques

Can classify solution approaches in to two categories:

- Data-based approaches-try to look at only a subset of the data such as sketch-based, sampling based etc.
- Task-based approaches-they are more focused on algorithmic changes in order to cope-up with the problems mentioned previously.

Contd...

Technique	Definition	Pros	Cons
Sampling	Choosing a data subset for analysis	Error Bounds Guaranteed	Poor for anomaly detection
Load Shedding	Ignoring a chunk of data	Efficient for queries	Very poor for anomaly detection
Sketching	Random projection on feature set	Extremely Efficient	May ignore Relevant features
Synopsis Structure	Quick Transformation	Analysis Task Independent	Not sufficient for very fast stream
Aggregation	Compiling summary statistics	Analysis Task Independent	May ignore Relevant features

Figure: Data based techniques [Aggarwal, 2007]

Contd...

Technique	Definition	Pros	Cons
Approximation Algorithms	Algorithms with Error Bounds	Efficient	Resource adaptivity with data rates not always possible
Sliding Window	Analyzing most recent streams	General	Ignores part of stream
Algorithm Output Granularity	Highly Resource aware technique with memory and fluctuating data rates	General	Cost overhead of resource aware component

Figure: Task based techniques [Aggarwal, 2007]

Classification Techniques

Classification techniques are usually based on the either data or task based. We will look at some classic and recent algorithms from both the categories. In particular, we will look at:

- Ensemble based classifier for handling data streams [Zhang et al., 2019]
- Decision tree based classifier called *Very Fast Decision Tree* (VFDT)

Re-sample based Ensemble Framework for Drifting Imbalanced Data Streams [Zhang et al., 2019]

Key Ideas:

- Two classifiers-one static and D dynamic ensemble-based classifier are proposed to handle gradual and sudden concept drifts respectively.
- Reinforcement-learning based weight adjustment procedure is introduced for increasing/decreasing weights of base classifier.
- Time-decayed weights for dynamic classifiers for focusing more recent examples.
- Uses buffer to re-sample and store instances from the minority classes for handling imbalance issue.

Contd...

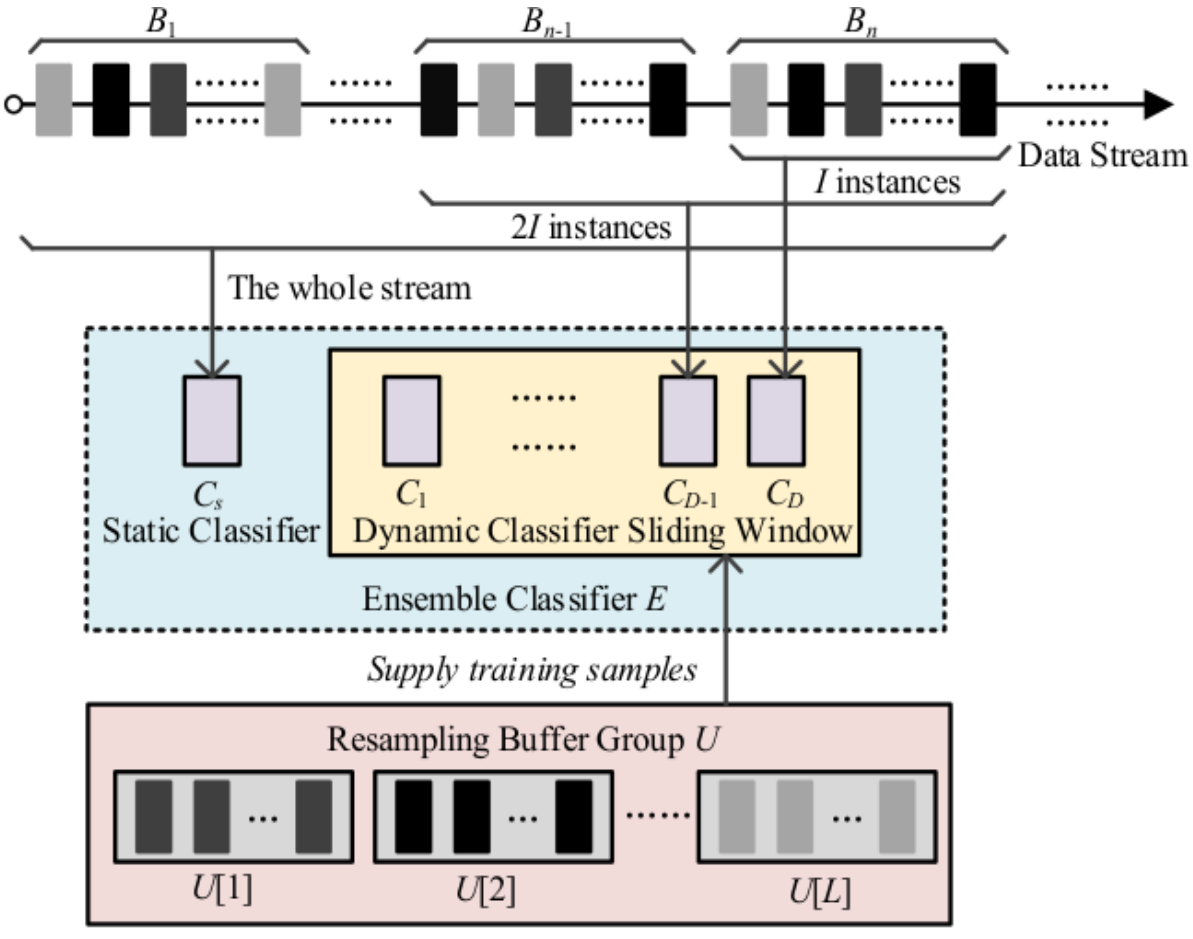

Figure: Resample based ensemble classifier for data stream. Fig. from [Zhang et al., 2019]

Contd...

Major Differences

Dynamic classifiers are created from static classifiers except they look at only a chunk of the data stream and they differ from static classifier in the weight adjustment strategy (see next slide)

Weight Adjustment Strategy

RE-DI algorithm proposed in [Zhang et al., 2019] does weight updates of static and dynamic classifers in two ways:

- It decreases weights of dynamic classifiers so that they can focus more on recent examples like so

$$w^s = \frac{1}{2}$$
$$w^d = w^d(1 - \frac{1}{D})(d = 1, \ldots, D-1)$$
$$w^D = \frac{1}{D}$$

- For handling class imbalance probelm, it employs so called reinforcement weight update strategy as follows:
  - if static and dynamic classifiers made mistake for *minority class* data point x, decrease weight by $1/D$ otherwise increase it by $1/D$.

Prediction Rule

RE-DI algorithm uses the following procedure to make a prediction:

$$f_l^E(x) = w^s f_l^{C_s}(x) + \sum_{d=1}^{D} w^d f_l^{C_d}(x)$$

where $f_l^E(x)$ is the ensemble prediction that instance x has class label l.

Very Fast Decision Tree (VFDT) Algorithm [Domingos and Hulten, 2000]

Key Ideas:

- It's fast incremental learning algorithm
- Needs a single pass over data
- Does not store examples in main memory
- It's *Anytime* algorithm which means you can stop the algorithm anytime with the same *performance guarantee*.
- The great thing about VFDT is that the tree produced by the algorithm is *asymptotically* similar to the tree produced by the batch learner.

OK. How VFDT works?

The working principle of VFDT is as follows:

1. It builds Hoeffding tree iteratively as it sees more examples.
2. It starts with the root node and picks an attribute to split based on usual node split criterion such as *Entropy*, *Information Gain*, *Gini Index*.
3. It does not split the node until the node has seen at least n examples so that with probability at least $1 - \delta$, difference in node's heuristic measure G is ϵ where ϵ is given by

$$\epsilon = \sqrt{\frac{R^2 \log(1/\delta)}{2n}}$$

Where R is the range of the attribute. Above equation is also called Hoeffding bound and whence Hoeffding tree comes from.

Bibliography I

Aggarwal, C. C. (2007).

Data streams: models and algorithms, volume 31.
Springer Science & Business Media.

Domingos, P. and Hulten, G. (2000).

Mining high-speed data streams.
In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 71–80.

Zhang, H., Liu, W., Wang, S., Shan, J., and Liu, Q. (2019).

Resample-based ensemble framework for drifting imbalanced data streams.
IEEE Access, 7:65103–65115.