International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Implementation of Very Fast Decision Rules for Classification in Data Streams

Vaddadi R V S Prasad

M-Tech, Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, Andhra Pradesh

Abstract: ***In this work huge amount of continuous and rapid data records are considered to solve the stream mining problems such as extracting knowledge structures by using several algorithms. One of the most interpretable and flexible model for predictive data mining are decision rules. Few algorithms are proposed in our work to have a clear idea on the decision rules. One of it is the VFDR algorithm and its proposed versions. These versions are one pass and any time algorithms. It works online and produces ordered or unordered rule sets. An adaptive extension is enabled to detect changes and quickly adapt the decision model. In this extension we monitor the evolution performance metric to detect concept drift. This explicit change detection mechanism provides useful information about the dynamics of process generating data.***

Keywords: Stream mining, decision rules, extracting knowledge, vfdr algorithm

1. Introduction

Now-a-days, with increasing importance of the online based information we obtain more and more useful data for every one minute. So it is very essential for the systems to find interesting patterns, discovering anomalies, detecting spams from the huge amount of data. It brings new challenges for machine learning which helps to process the information and train the systems in discovering frauds, predicting the future data and grouping similar data.

Now in many real world applications and ubiquitous mining the data is represented in the form of streams. Data streams refer to a process where instances arrive continuously and possibly infinitely over time. It is unmanageable to process the data streams as they are very quick and continuous in nature. So we might not consider all the data at a time, we need sequential access rather than random access. There is a need to emerge a new extension to the offline static mining methods to handle this type of data. The improved version need to posses the characteristic such as to produce the model for the given input. The model must be defined at any point of time by scanning the data for only once.

Data streams possess a special behavior which is known as concept drift, the data changes its nature with time under unknown dynamics. The functional mapping between the attribute and the classes change by the concept drift. The model that was built on the previous observations is no longer useful, so the new model should be able to detect the changes and adapt to the current input in quick time else, it may cause adverse effect on quality of the system. Faster adaptation is an advantage to stream mining problems[1].

Classification, the most important technique of Machine learning which is useful to classify and predict the data from large data bases. Decision trees are one of the most popular tools of classification. They are hierarchical in structure in which nodes represent decisions and leaves represent class labels[1]. They can be easily interpreted because of their comprehensive visualization and also provide high degree of predictive capabilities. Thus the path from the root to leaves can be rewritten into IF-Then decision rules. These rules capture the main characteristic of the decision problem. Each unordered rule can be handled independently of others in the rule set. Decision rules can be induced either directly from the data by using rule learning algorithms or by retrieving rules from the decision trees built on the data[2].

In dealing with the evolved data, it is very important to observe changes in the model. In data stream-classification the best used models to adapt the drift are Hoeffding trees[2]. These are incremental algorithms which automatically detect the changes and adapt to concept drift by just expanding the tree by implicit adaptation method. However, this process is rather slow. Faster adaptation might be achieved by explicit adaptation method but, it is computationally expensive[3]. So, the implicit adaptation method remains in the tree. In this article we present VFDR learning algorithms to classify the data streams and adaptive extension how to learn rules when the data evolves its nature. We organized rest of the article as follows. Section 2 Discusses about the related work and our proposed method is introduced in Section 3. Results are presented in section 4.

2. Related Work

2.1 Stream Classification Algorithms

We have many algorithms to classify the data streams among them very fast decision tree(VFDT) presented by Domingos and Hulten is a popular one. It is a classifier used to classify the stationary data streams. In this model when an input traverses from the root to the leaf of the tree according to the attribute values in the node, It is then classified by the class label presented in the leaf. If the sufficient statistics needed by the heuristic function is satisfied then it evaluates the merit of the split tests on attribute values. When an input is traversed till the leaf the sufficient statistics are updated. Further, the leaf is replaced with test node after acquiring satisfied hoeffding bound.

The value of test node is derived from the split evaluation function for all the values in the available attributes. This work had acquired lot of attention from the community of mining scenario. Different extensions and improvements are done to VFDT[2].

Hulten(2001) presented a improved version of VFDT known as CVFDT, in order to handle the non-stationary data streams. In this model the data is changing under the influence of several factors. The split that was previously done is no longer accurate. In this case CVFDT starts learning a sub tree of new best attribute split. This CVFDT algorithm keeps the model up to date from the concept drift.

2.2 Rule Learning

Rules can be learnt in two ways: 1.Extracting the rules directly from the data by using the algorithms like RIPPER etc 2. Building a decision tree from the given data and then learning rules from it. Any decision tree can easily be transformed into collection of rules.

Number of rules should be as many as leaves present in the tree. This process generates rules with same complexity of the tree [3]. C4.5 rules uses an optimization procedure to simplify the condition of the rule by removing unnecessary literals of the rule which are not effecting the accuracy of the rule. And then removing duplicates of the rules in the set. These rule sets are more accurate and simpler than the initial rules derived from the tree.

In indirect method, RIPPER solves the problem in one vs all fashion[4]. The classes are ordered according to priorities. Initially it sets one class as positive and all the other classes are set to negative. The rules are generated for the positive examples and then positive examples are removed from the set. Then the algorithm moves to the next class in greedy fashion in every iteration.

2.3 Adaptive Methods

There are many approaches to handle this adaptive method to deal with concept drift such as sliding window example weights and the ADWIN algorithm. But in this article we deal with explicit method which is computationally an expensive task. It did not acquire much attention from the stream mining community because of its complicated process. It explicitly implies the rules which are inconsistent and those rules are removed to avoid adverse effect on the quality of the model[6].

3. Methodology

In this work we implement very fast decision rule algorithms to classify the data streams. It is a online classifier which is able to provide the model while scanning the data only once. Initially we discuss about the base version first and then its improved versions in later.

3.1 VFDR-Base Algorithm

It is designed to classify the high speed data streams, it learns ordered and unordered rule sets[1].

- **Rule Learning Phase**

A decision rule is in the implication of the form A→C Here A is the conjunction of the literals C is the constant as in many other rule learning algorithms. But in VFDR the C part of the rule is the class label assigned by the majority class or NB.

To learn the rules in VFDR the algorithm begins with an empty rule set (RS) and the default rule L is initialized to NULL. Here L is the data structure that stores an integer that stores the number of examples covered by the rule ; P(C_k) that stores the number of examples that observed for the particular class and updates the sufficient statistics that when to expand the rule and how to classify the test instances[1].

If all the lierals are true for the given labeled example, then it is set to be covered by the rule. Then it updates the statistics of L_r which is used to expand the rule with highest gain measure of the literal.

After how many examples certain rule was to be expand or new rule are to be induced was determined by the Hoeffding bound (Ɛ). It is not efficient to check with every incoming example. Therefore it is done only after N_{min} examples[2].

The rule expansion or new rule inducing was done by computing split evaluation function for every value v_j in the every attribute x_i. Then we found best literal and second best literal named for g_{best} and g_{2best} respectively and are used for Hoeffding bound. If these two literals is better than the given confidence i.e (g_{best} - g_{2best} > ϵ) then the rule is expanded with condition x_a=v_j and the class label was given according to the majority class of x_a=v_j or NB.

In this we learn ordered and unordered rule sets in which the every labeled example updates statistics of the first rule that covers it. In unordered every labeled example updates statistics of all the rules that cover it. If it is not covered by any of the rule in the setthen default rule is updated.

- **Classification Strategies**

When an unlabelled or test example covers the rule then the example is classified according to the information stored in L_r of that rule. Simply it is done by the majority class distribution i.e P(C_k). In this strategy L_r uses very small part of the available information.

By applying Naïve bayes in VFDR we get more sophiscated results when compared to majority class. It is because the learned rule set may not contain the sufficient rules but using NB in the VFDR provides highly interpretable results of class prediction.

VFDR-BASE: Rule learning Algorithm

input : *S*: Stream of examples
N_{min}: Minimum number of examples
ordered set: boolean flag

output: *RS*: Set of Decision Rules

begin
Let *RS* ← {}
Let *default rule L* ← ø
foreach *example* $(x, y_k) \in S$ **do**
foreach *Rule r* ∈ *RS* **do**
if *r covers the example* **then**
Update sufficient statistics of Rule *r*
if *Number of examples in* $L_r > N_{min}$
then
r ← *ExpandRule*(*r*)
if *ordered set* **then**
BREAK;
if *none of the rules in RS trigger* **then**
Update sufficient statistics of the empty rule
if *Number of examples in L* > N_{min} **then**
RS ← *RS* ∪ ExpandRule(*default rule*)

VFDR: Rule Expansion Algorithm

Input : *r*: One Rule
δ : User defined confidence level
Output: *r*: Expanded Rule
Begin
Compute $\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2N}}$ (Hoeffding Bound)
Find the best g_{best} and second best g_{2best} attribute merit
If ($g_{best} - g_{2best} > \epsilon$) **then**
Extend r with a new condition based on the best attribute $x_a = v_j$
Reintilialize sufficient statistics of L_r
r ← r ∪ $\{x_a = v_j\}$
return r

3.2 One vs All Rule Learning

This is the modified version of basic algorithm VFDR for learning from multi class labels. In this version, one class was set to positive and all the other remaining classes were set to negative[4]. The rules are learnt according to each class respectively, when it is set to positive. The rule expansion was done for each class C belongs to C_r where C_r means set of classes observed at rule r. The number of rules induced from one rule is at most $|C_r|$.

The VFDR-OA adopted FOIL measure to select the new condition .For each attribute X_i we compute this measure to find the best literal which maximizes the accuracy of the rule.

$$\text{Gain}(r',r) = s \times \left(\log_2 \frac{N'+}{N'} - \log_2 \frac{N+}{N}\right)$$

where *N* is the number of examples covered by *r* and *N*+ is the number of positive examples in them, *N*'+ and *N*_ represent the same for *r'*, and *s* is the number of true positives in *r* that are still true positives in *r'*, which in this case corresponds to *N*' +[4]. We are interested only in positive gain, therefore we consider the minimum of the gain function as zero and the maximum for a given rule is $N_+ \times \left(-\log_2 \frac{N_+}{N}\right)$.
We can then normalize the positive gain as:

$$\text{Gain Norm } (r',r) = \frac{\text{Gain}(r',r)}{N_+ \times \left(-\log_2 \frac{N+}{N}\right)}$$

While in the case of expansion of rule which already have conditions i.e, it is non-default rule and the procedure is as follows. In the expansion of ordered rule the rule of which is only positive class was expanded. There is a different scenario in expanding unordered rule compared to the positive class expansion. It also computes the best literals of the other classes with one call of Expand rule.

The additional expansions are allowed only when the rule has already been expanded with the positive class. The advantage of this algorithm is it can learn more sophisticated and complex rules from few examples[4]. By this we can achieve high accuracy of large data sets. The search for the best and second best values considers the value of *GainNorm* for a given class. If g_{best} is the true best gain measure, i.e., satisfies condition $g_{best} - g_{2best} > \epsilon$ **for a given class** C_k the rule is expanded with condition $X_a = V_j \rightarrow C_k$

3.2.1 Multiple Rule Sets

The main point of this version is to have separate rule sets for each class in the data. It was done on the basis of the idea of VFDR-OA. But there is slight modification in the learning and prediction phase when compared to previous version. The main aim of this modification is to have rules for every class present in the data.

- **Learning Phase**

In this we have rule subsets for each class and a default rule. This differentiates that one class has rule subset which is considered positive; all the other class labels were negative. £r of the each rule contains the information about two class problem. The rule expands only when the rule subset class indicates the expansion. It adopts the FOIL gain measure to get the best literal to expand the rule. When a new example arrives it updates the statistics of the rules that cover it. If the class of the example is same as the subset class the example is used or else it is not used.

- **Prediction Phase**

Generally the predictions were done by using majority class or NB but in this it was done with slight modifications. That is when a class predicted by MC or NB is not a subset class then it determines that the rule is not covered in the test example. The rule subset itself denotes that they are predictions of its subset class supported by weight given by the rules covered by the examples. The final prediction is done by using weight given by the rules that covered the examples. Subset weights are determined by the WeightMax and FirstHit. If there was no rule that covers the example and

predict the class of its subset then default rule provides the prediction based on MC or NB.

3.3 Adaptive Extension

The main characteristic of the data streams is its evolving nature. The model which we have built on the previous observations were no longer useful. It is important to detect the changes in the model quickly and make changes to it is an important issue.

In order to detect the changes in this model we adopt the explicit change detection mechanism to detect the concept drift in the data and remove the rules which are outdated making negative influence of the results[5].

Explicit detection mechanism employs an Statistical Process Control(SPC) method to handle this evolving data[6]. It has two registers p min and s min whenever a rule covers a labeled example. It makes a prediction and updates error value by using these two registers .Whenever a new example arrives it updates these values.

$$p_i + s_i < p_{min} + s_{min}$$

This learning process can give any one of the following states: In Control Warning Out-of-Control.

$$p_i + s_i \geq p_{min} + \alpha \cdot s_{min}$$

Where α=2 changes to Warning state and when α=3 implies it changed to Out of control state.

When the learning process is in the Warning state the learning classifier waits until the following state becomes In-Control state. Whereas in the Out-of-Control state, the particular rule implies the negative influence on the quality of the model. Therefore rule was removed from the rule-set. Thus, it makes the rule-set consistent with non-stationary data and avoids computational task to build the model from the scratch[6].

Dataset	VFDR-Base-UN	VFDR-Base-OR	VFDR-MS-UN	VFDR-MS-OR	VFDR-OA-UN	VFDR-OA-OR
Hyperplane	14.72 (0.07)	14.95 (0.12)	15.65 (0.54)	14.62 (0.19)	**13.25** (0.59)	16.06 (0.63)
LED	41.67 (3.11)	26.46 (0.05)	26.60 (0.14)	26.61 (0.19)	**26.22** (0.09)	27.43 (0.09)
SEA	15.33 (0.04)	15.54 (0.09)	14.75 (0.33)	13.77 (0.19)	**12.10** (0.17)	13.48 (0.40)
RBF	22.23 (1.88)	24.42 (0.23)	19.56 (3.29)	22.82 (4.86)	**18.98** (3.52)	22.20 (4.28)
Waveform	19.09 (0.31)	19.84 (0.09)	16.49 (0.19)	17.35 (0.20)	**16.35** (0.25)	20.63 (0.79)
Average rank	4.4	4.6	3.2	3.4	1	4.4

Figure 1: Prequential error of the Unordered and ordered VFDR classifiers

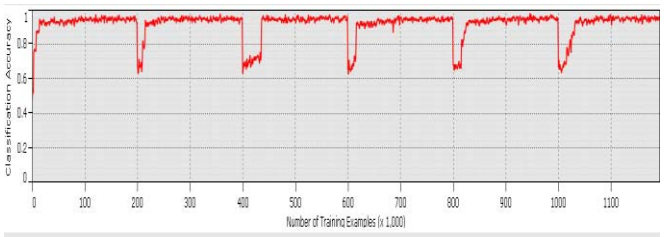

Figure 2: Accuracy of VFDR-OA classifier.

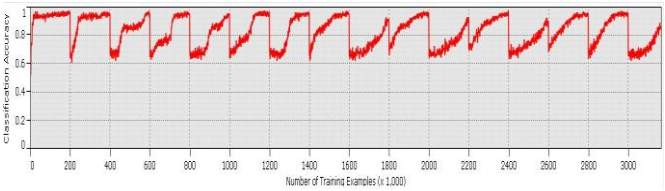

Figure 3: Accuracy of AVFDR-OA Classifier.

4. Conclusion

In this paper we present VFDR Base algorithm and proposed versions in learning decision rules to classify the data streams. This is an online classifier and it is able to produce the model while scanning the data for only once. In the proposed versions the complex rules are learnt for each class to improve the accuracy of the system by only just covering few training examples. To detect the evolving nature of the data we adopted explicit detection method i.e, Statistical Process Control(SPC) as an adaptive extension. By this we can simply find and remove the rules which are outdated to the current concept. This method employs high performance on this evolving data and adopts faster adaptation while compared to other methods.

References

[1] Very fast decision rules for classification in Data streams - Joao Gama and Petr Kosina
[2] Learning decision rules from Data Streams – Joao Gama and Petr Kosina.
[3] Mining high-speed Data Streams – Pedro Domingos and Geoff Hulten.
[4] Very fast decision rules for Multi-class problems – Petr Kosina and Joao Gama
[5] Handling time changing data with Adaptive Very Fast Decision Rules – Petr Kosina and Joao Gama.
[6] Learning with Drift Detection – Joao Gama Pedro Medas Gladys Castillo & Pedro Rodrigues.