Adaptive Classification Algorithm for Concept Drifting Electricity Pricing Data Streams

Pramod D. Patil
Research Scholar
Department of Computer Engineering
College of Engg. Pune, University of Pune

Parag Kulkarni, PhD.
Research Guide
Department of Computer Engineering
College of Engg. Pune, University of Pune

ABSTRACT

Electricity is the main observation in our daily life. There are many parameters or a factors on which the electricity load is depends on, knowable load factors such as whether conditions, temporal factors, and customer characteristics etc. Daily peak load is an important factor in the planning the production and pricing of electricity. In a simple terms, it is essential to get the knowledge of the local system demand will be on the next minutes, hours and days so that the generators with various startup times, startup cost can be changes as per the requirement and knowledge gain from the previous data collected. This paper is intended for industry/ organization to optimize Electricity usage. Energy consumption and pricing analysis is a primary area in power systems planning and management. Recent developments in energy market deregulation and provision of sustainable energy have contributed to increase interest in this area. The prices are not fixed and are affected by demand and supply of the market. The prices in electricity market can be set every five minutes.

With this motivation, an algorithm is proposed for efficient Classification of concept Drifting Electricity pricing data streams. Thus, it is a challenge to learn from concept drifting data streams. In proposed algorithm, a decision tree is built incrementally and also used to develop training set based on these methods, in order to improve the accuracy of classification and prediction models under concept drift. A base learner is adaptive, a decision tree can have its nodes included and deleted dynamically. Adaptivity can be achieved by manipulating training data (instance selection), instead of taking all training history, take a number of the latest instances (training window). The new proposed algorithms detect change faster, without increasing the rate of false positives. Extensive studies on both synthetic and real-world data demonstrate that proposed algorithm outperforms well compared to several state-of-the-art online algorithms.

In this paper we have compared electricity datasets with three algorithms to find out the algorithms efficiency on type of dataset. This data is again tested for error value for a particular number of iteration. The experimentation is conducted. The experimental evaluation produced satisfactory results.

Keywords—Decision trees, Data Streams, Incremental learning

1. INTRODUCTION

Global population is expected to grow by 1% per annum on average, from an estimated 6.4 billion in 2004 to 8.1 billion in 2030. The demand for energy is growing at an average of 1.6% per annum. Global Energy Demand is projected to increase by 53% from 2004 to 2030. 70% of increase in primary energy demand during this period comes from developing countries. A result of the economy and population growth in these countries, which shifts the centre of gravity of global energy demand. Power generation accounts for 47% of increase in global energy demand by 2030. Among all major end-use energy sources, electricity is projected to grow most rapidly by 2.6% per year as against the 1.6% growth of primary energy demand. Meeting the world’s growing hunger for energy requires massive investment in the energy supply infrastructure. WEO-2006 calls for a $20 trillion (in year 2005) investment over 2005-2030. Power Sector accounts for 56% of total investment. In many applications, learning algorithms act in dynamically where the data flow is continuous. If the process is not stationary the target concept could change over time. In real world problems, change detection is relevant. These include user modeling, monitoring in the bio-medicine and industrial processes, fault detection and diagnosis, safety of complex system etc.

The electricity market dataset used in this experiment was first described by M. Harris[4]. The dataset contains 45312 instances. Each example of the dataset refers to a period of 30 minutes i.e there are 48 instances for each time period of one day. Each example on the dataset has 5 fields the day of week, the timestamp, the NSW electricity demand, the vic electricity demand, the schedule electricity transfer between states and the class label. The class label identifies the change of the price related to a moving average of the last 24 Hours. The class level only reflects deviation of the price on a one day average and removes the impact of longer terms price trends. The interest of this dataset is that it is a real world dataset. Using artificial dataset allow us to control relevant parameters to evaluate drift detection algorithm. For example, we can measure how fast the detection algorithm reacts to drift. Evaluation methodology changes occur over time. Drift detection algorithm assume that data is sequential.

The proposed CDDT algorithm makes several modifications to the Hoeffding tree algorithm to improve both speed and memory utilization. The modifications include breaking near-ties during attribute selection more aggressively, computing the G function after a number of training examples, deactivating the least promising leaves whenever memory is running low, dropping poor splitting attributes, and improving the initialization method. It works well on stream data and also compares extremely well to traditional classifiers in both speed and accuracy to adapt to concept-drifting data streams, this algorithm we can further changed by using the concept drift in the used data streams. Decision Tree by considering the drift in the data streams, this also uses a dynamic training window approach. However, it does not construct a new model from scratch each time. Rather, it updates statistics at the nodes by incrementing the counts associated with new examples and decrementing the counts associated with old ones. Therefore, if there is a concept drift, some nodes may no longer pass the Hoeffding bound. When this happens, an alternate subtree will be grown, with the new best splitting attribute at the root. As new examples stream in, the alternate subtree will continue to develop, without yet being used for classification. Once the alternate subtree becomes more accurate than the existing one, the old sub tree is replaced.

The rest of this paper is organized as follows. We start with an overview of related work in Section 2. Section 3 provides general framework of incremental learning based on decision tree before we present our Adaptive classification algorithm of CDDT in Section 4. Section 5 provides the experimental studies and Section 6 summarizes our results and future work.

2. LITERATURE REVIEW

Standard decision tree learners such as ID3, C4.5, and CART assume that all training examples can be stored simultaneously in main memory, and are thus severely limited in the number of examples they can learn from. In particular, they are not applicable to data streams, where potentially there is no bound on the number of examples and these arrive sequentially.

GEMM and FOCUS [6] are used for building decision tree and frequent item sets with concept drifting data streams in incremental models. But it is time consuming and costly learning.

OLIN [6] uses info-fuzzy techniques for building a tree-like classification model. It is used for dynamic updates. But it is also time consuming and costly learning and storage memory problem.

VFDT [1] based on incremental building of decision tree with high speed and need less memory space. It is non-adaptable to concept drift. It is costly learning algorithm.

LWClass[2,13] based on classes weights with high speed and less memory space. It is non-adaptable to concept drift.

CDM [3] is a combination of decision tree and Bayes network. It is used for suitable factor to measure distance between events.

Ensemble based [12] Classification using combination of different classifiers. It is single pass and concept drift adoption and high accurate. Disadvantages of this algorithm are low speed, storage memory problem and time consuming and costly learning.

SCALLOP [3] Scalable classification for numerical data streams with dynamic updates.

Domingos [1,7] proposed the Hoeffding tree as an incremental, anytime decision tree induction algorithm that is capable of learning from data streams, assuming that the distribution generating examples does not change over time. Hoeffding trees exploit the fact that a small sample can often suffice to choose a splitting attribute. This idea is supported by the Hoeffding bound, which quantifies the number of observations (in our case, examples) needed to estimate some statistics within a prescribed precision (in our case, the goodness of an attribute). A theoretically appealing feature of Hoeffding Trees not shared by other incremental decision tree learners is that it has sound guarantees of performance. Using the Hoeffding bound one can show that its output is asymptotically nearly identical to that of a non-incremental learner using infinitely many examples. The given algorithm is an extension of the Hoeffding Tree to evolving data streams, but does not exhibit theoretical guarantees.

Contrary to the aforementioned algorithms, we propose a efficient and Adaptive classification algorithm of CDDT for data streams with concept drifts. CDDT provides the following three main characteristics. Firstly, a decision tree is generated incrementally. Secondly, potential concept drifts are detected corresponding to the deviations of classification in the history concept and new ones. Lastly, an approach of bottom-up search is utilized to trace all drifting leaves. CDDT achieves better performances compared to several known classification algorithms for concept drifting data streams based on single and ensemble models.

3. INCREMENTAL LEARNING

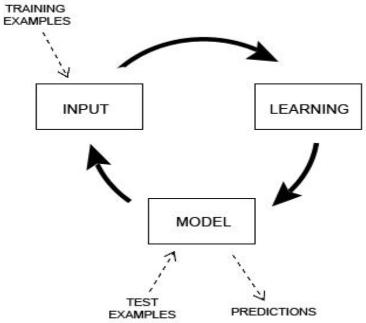

Fig. 1 Learning Framework

A. Hoeffding Trees [9]

A classification problem is defined as:

- N is a set of training examples of the form (x, y)
- x is a vector of d attributes
- y is a discrete class label
- Goal: To produce from the examples a model y=f(x) that predict class y for future examples x with high accuracy.

B. Hoeffding Bound [1,11]

- Consider a random variable a whole range is R
- Suppose we have n observations of a
- Mean: $\bar{a}$
- Hoeffding bound states:

-With probability 1-δ, the true mean of a is at least, where

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

- Let G (Xi) be the heuristic measure used to choose test attributes (e.g. Information Gain, Gini Index)
- Xa: the attribute with the highest attribute evaluation value after seeing n examples.
- Xb: the attribute with the second highest split evaluation function value after seeing n examples.

•Given a desired δ, if after Seeing n examples at a node, $\Delta\bar{G} = \bar{G}(X_a) - \bar{G}(X_b) > \varepsilon$

-Hoeffding bound guarantees $\Delta G >= \Delta\bar{G} - \varepsilon > 0$ with probability 1-δ.

-This node can be split using Xa; the succeeding examples will be passed to the new leaves.

4. ALGORITHM DESIGN AND IMPLEMENTATIONS

Here we are going to compare three algorithms with each other and analysis all results from these three algorithms

4.1 Hoeffding Algorithm

Inputs:

S is a sequence of examples,

X is a set of discrete attributes,

G (.) is a split evaluation function,

δ is one minus the desired probability of choosing the correct attribute at any given node.

Output: H T is a decision tree.

```
• Procedure HoeffdingTree (S, X, G, δ)
• Let H T be a tree with a single leaf l1 (the root).
        Let X1 = X ∪ {X∅}.
• Let $\bar{G}_1$ (X∅) be the $\bar{G}$ obtained by predicting the most frequent class in S.
• For each class yk
        -For each value xij of each attribute Xi ∈ X
        -Let nijk (l) = 0.
• For each example (x, yk) in S
        -Sort (x, y) into a leaf l using HT.
        -For each $x_{ij}$ in x such that Xi ∈ Xl
        -Increment $n_{ijk}$ (l).
• Label l with the majority class among the examples seen so far at l.
• If the examples seen so far at l are not all of the same class, the
        -Compute $\bar{G}_l$ ∈ (Xi) for each attribute Xi ∈ $X_l$ - {X∅} using the counts nijk (l)
        -Let Xa be the attribute with highest Gl.
        -Let Xb be the attribute with second-highest Gl. Compute using Equation 1.
        -If $\bar{G}$l (Xa) − $\bar{G}$l (Xb) > and Xa = X∅, then
        -Replace l by an internal node that splits on Xa.
        -For each branch of the split
• Add a new leaf lm, and let Xm = X − {Xa}.
• Let $\bar{G}$m (X∅) be the G obtained by predicting the most frequent class at lm.
• For each class yk and each value xij of each attribute Xi ∈ Xm − {X∅}
• Let $n_{ijk}$ (lm) = 0.
• Return HT
```

4.2 Very Fast Decision Tree Algorithm

```
VFDT (stream, δ)
{
        -Let HT be a tree with a single leaf (root)
        -Init count nijk at root to 0
        -For each example (x,y) in a stream
        -Do VFDTGrow ((x,y), HT,δ)
}
VFDTGrow ((x, y), HT, δ)
{
        -Sort (x,y) to leaf l using HT
        -Updates count $n_{ijk}$ at leaf l
        -If example seen so far at l are not all of the same class then compute $\bar{G}$ for each attribute
        -If $\bar{G}$ (best attribute) – $\bar{G}$ ($2^{nd}$ best attribute) >
```

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

```
                -Then split leaf on the best attribute
                -For each branch
                -Do start new leaf and initialize count
}
```

4.3 Concept drift algorithm

Dynamic construction of the tree from the data streams i.e. by considering the concept drift in the data we can built the tree from the dynamic result so that accuracy of the decision tree can be maximum.

• Alternate trees for each node in HT start as empty.

• Process examples from the stream indefinitely. For each example (x, y),

-Pass (x, y) down to a set of leaves using HT and all alternate trees of the nodes (x, y) passes through.

-Add (x, y) to the sliding window of examples.

-**Remove and forget the effect of the oldest examples**, if the sliding window overflows. (**Go to Step1**)

-**CDDTIncrease (Go to 2)**

-**Split** if f examples seen since last checking of alternate trees (**Go to 3**)

• Return HT.

Step1.

-Maintain the sufficient statistics at every node in HT to monitor the validity of its previous decisions.

-VFDT only maintains such statistics at leaves.

-HT might have grown or changed since the example was initially incorporated.

-Assigned each node a unique, monotonically increasing ID as they are created.

-**Step1** (HT, example, maxID)

-For each node reached by the old example with node ID no larger than the max leave ID the example reaches

-Decrement the corresponding statistics at the node.

-For each alternate tree Talt of the node, forget (Talt, example, maxID).

Step2.

-For each node reached by the example in HT,

-Increment the corresponding statistics at the node.

-For each alternate tree Talt of the node,

- **Step2**

-If enough examples seen at the leaf in HT which the example reaches,

-Choose the attribute that has the highest average value of the attribute evaluation measure (information gain or gini index).

-If the best attribute is not the "null" attribute, create a node for each possible value of this attribute

Step3.

-Periodically scans the internal nodes of HT.

-Start a new alternate tree when a new winning attribute is found.

-Tighter criteria to avoid excessive alternate tree creation.

-Limit the total number of alternate trees.

5. RESULT ANALYSIS: ELECTRICITY DATASET

5.1 Hoeffding Analysis

Following are the results we are getting from the Hoeffding algorithm

i) Example Details:
-Number of Classes: 2
-Number of Examples: 600
-Majority Class Label: DOWN

ii) Class Distribution

Classes	Label	Proportion
0	UP	44.5%
1	DOWN	55.5%

iii) Tree Information
-Number of Nodes: 1761

-Number of Leaf Nodes: 857

-Number of Levels: 4

iv) Classification Details:
-Number of Test Observation: 100
-Misclassified: 55%
-Classified: 45%

The Error rate as per the calculation is shown here. In the 100 examples we are getting the 55% misclassified and 45% classified. Hence Error Percentage is the 55%

v) Error Rate:

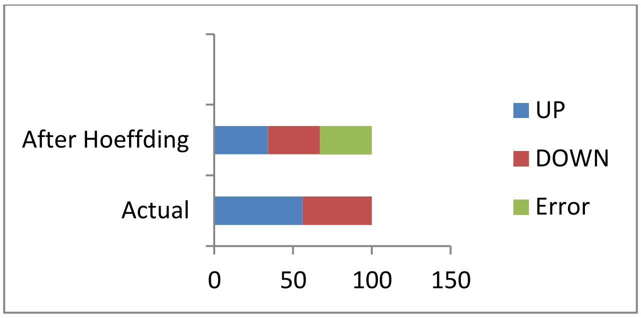

Fig.2 Error Rate: Hoeffding Tree

5.2 VFDT Analysis

Following are the results we are getting from the VFDT algorithm

i) Example Details
-Number of Classes: 2
-Number of Examples: 600
-Majority Class Label: DOWN

ii) Class Distribution

Classes	Label	Proportion
0	UP	44.5%
1	DOWN	55.5%

iii) Tree Information
-Number of Nodes: 97

-Number of Leaf Nodes: 48

-Number of Levels: 2

iv) Classification Details:
-Number of Test Observation: 100
-Misclassified: 45%
-Classified: 55%

The Error rate as per the calculation is shown here. In the 100 examples we are getting the 45% misclassified and 55% classified. Hence Error Percentage is the 45%

v) Error Rate

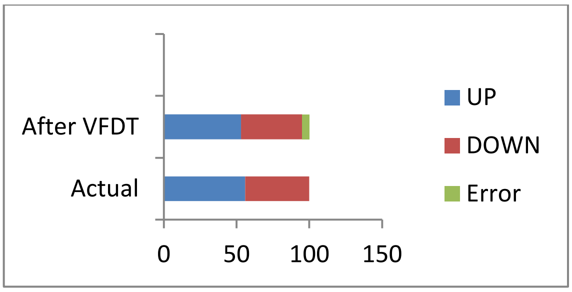

Fig.3 Error Rate: VFDT

5.3 Concept Drift Analysis

Results of the CDDT algorithm are as below

i) Example Details
-Number of Classes: 2
-Number of Examples: 600
-Majority Class Label: DOWN

ii) Class Distribution

Classes	Label	Proportion
0	UP	49.4%
1	DOWN	50.6%

iii) Tree Information
-Number of Nodes: 97

-Number of Leaf Nodes: 48

-Number of Levels: 2

iv) Classification Details:
-Number of Test Observation: 400
-Misclassified: 29.75%
-Classified: 70.25%

The Error rate as per the calculation is shown here. In the 400 examples we are getting the 29.75% misclassified and 70.25% classified. Hence Error Percentage is the 29.75%

v) Error Rate

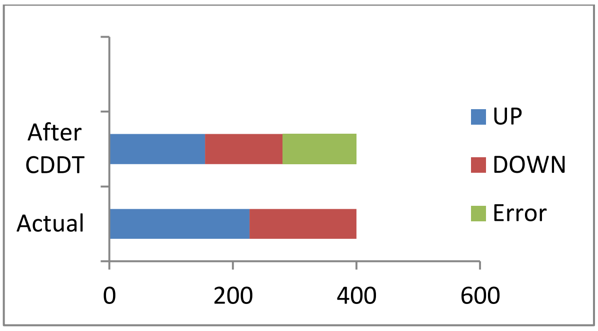

Fig.3 Error Rate: CDDT

5.4 Comparison between all algorithms

Final Result Analysis by considering the all three explained algorithms is as follows.

i) Require memory O (nodes * attributes * attribute values * classes).

ii) Running time O (Lc * attributes * attribute values * number of classes).

iii) Running Time: -VFDT: 12 Seconds -CDDT:43 Seconds

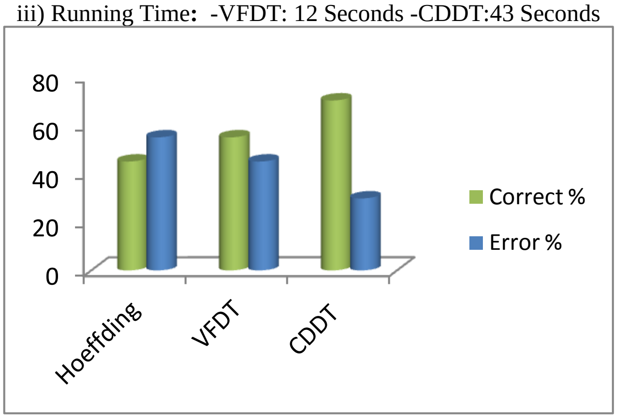

Fig.4 Error Rate: Comparison

5.5 COMPARISON BETWEEN DATASET AND ERROR RATES

Table 1Comparision Parameters

Dataset		Electricity
Records		Floats
Classes		Char
No. Of classes		2
Actual Records		45312
No. of Records		1000
Tested records		100
Hoeffding	Classified	45
	Misclassified	55
VFDT	Classified	55
	Misclassified	45
CDDT	Classified	71
	Misclassified	29
Error Rates	Hoeffding	55%
	VFDT	45%
	CDDT	29%
Time taken (In Seconds)	Hoeffding	10
	VFDT	12
	CDDT	43

6. CONCLUSION

This paper introduced Hoeffding trees, a method for learning from the high-volume data streams that are increasingly common. Hoeffding trees allow learning in very small constant time per example, and have strong guarantees of high asymptotic similarity to the corresponding batch trees. VFDT is a high-performance data mining system based on Hoeffding trees. Empirical studies show its effectiveness in taking advantage of massive numbers of examples. Error rates are going to minimized from Hoeffding Algorithm to CDDT algorithms. In the last section of we have shown the all parameters related to the dataset, error rates, records, test records and time taken for the evaluation of the decision tree. Experimental evaluations reveal that in comparison to several state-of-art methods, proposed algorithm is effective and efficient. An application of CDDT on a real-world Electricity pricing database has also shown promising results. Meanwhile, how to identify better discretization approaches to the numerical attribute values, how to reduce the overheads of space and how to predict unknown concepts in advance are still challenging and interesting issues for our future work.

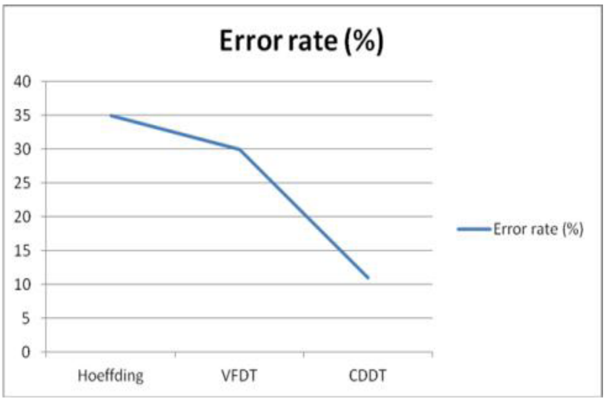

Fig. 5 Error Rate

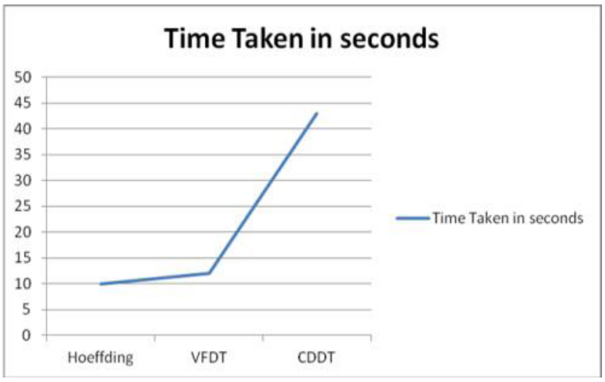

Fig 6. Run Time

7. ACKNOWLEGEMENTS

The project entitled " Use of GSM/GPRS for Residential Meter Monitoring" is sponsored by Pune University, India This research is supported by Pune University under BCUD Research Project Grants. We have successfully completed this project.

8. REFERENCES

[1] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In KDD '01: Proc. of the 7th ACM SIGKDD int. conf. on Knowledge discovery and data mining, pages 97–106. ACM, 2001

[2] R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. Intelligent Data Analysis, 8(3):281–300, 2004.

[3] N. Street and Y. Kim. A streaming ensemble algorithm (sea) for large scale classification. In KDD '01: Proc. of the 7th ACM SIGKDD int. conf. on Knowledge Discovery and Data Mining, pages 377–382. ACM, 2001.

[4] Michael Harries. Splice-2 comparative evaluation: Electricity pricing.Technical report, The University of South Wales, 1999.

[5] J. Gama, P. Medas, G. Castillo, and P. Rodrigues. Learning with drift detection. In SBIA Brazilian Symposium on Artificial Intelligence, pages 286–295, 2004.

[6] Mining Data streams : a Review, Vol. 34 No.2, June 2005

[7] Classification using streaming Random Forests, IEEE Transaction on Knowledge and Data Engineering,Vol.23 No.1, 2011

[8] Relevant Data Expansion for Learning Concept Drift from Sparsely Labeled Data, IEEE Transaction on Knowledge and Data Engineering,Vol.17 No.3, 2005

[9] Decision Tree for Uncertain Data, IEEE Transaction on Knowledge and Data Engineering, Vol. 23 No.1, 2011

[10] Top-Down Induction of Decision Tree Classifier – A Survey , IEEE Transaction on systems, man and cybernetics, Vol.35 No.4,2005

[11] Learning Decision Trees from Dynamic Data Streams, Journal of Universal Computer Sciense,Vol.11 No.8 , 2005

[12] Robust Ensemble Learning for Mining noisy data streams, ELSEVIER : Decision Support System, P. 469-479, 2010

[13] Mining Massive Data Streams, Journal on Machine Learning 2005