http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (24): 8469-8475, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.8469.8475

Learning Decision Trees from Time-changing Uncertain Data Streams

[1]Chunquan Liang, [1]Yang Zhang and [2]Shaojun Hu
[1]College of Mechanical and Electronic Engineering, Northwest A and F University, Yangling, 712100, Shaanxi, People Repbulic of China
[2]College of Information Engineering, Northwest A and F University, Yangling, 712100, Shaanxi, People Repbulic of China

Abstract: In this study, we study the problem of classifying uncertain data streams. Based on CVFDT algorithm, we proposed a novel algorithm, namely uCVFDTc, to learn very fast decision trees from uncertain data streams with concept drift. In training phase, the uCVFDTc algorithm uses Hoeffding bound theory to yield fast and reasonable decision trees. In classification phase, at tree leaves it uses Uncertain Naive Bayes (UNB) classifiers to improve classification performance. Experimental results showed that uCVFDTc had strong ability to learn from uncertain data streams and cope with concept drift; the use of UNB at tree leaves had improved the performance of uCVFDTc, especially the ability to handle concept drift.

Key words: Uncertain data streams, very fast decision tree, concept drift

INTRODUCTION

Data stream classification has been widely used in many applications, e.g., credit fraud detection, network intrusion detection and environmental monitoring (Mala and Dhanaseelan, 2011). In these applications, huge amount of streaming data is continuously generated at a high speed, along with changing concept. So the mining tasks on data streams require incremental, limited memory, any-time and concept adapting learning algorithms (Domingos and Hulten, 2000). Based on ensemble learning (Polikar, 2006) or decision tree (Quinlan, 1993), many of such algorithms have been proposed (Gama, 2012). The most famous one is concept Adapting Very Fast Decision Tree (CVFDT) algorithm (Hulten *et al*., 2001). Based on sliding window technique and Hoeffding bound theory (Hoeffding, 1963), CVFDT can learn decision trees online with bounded memory usage, high processing speed and detecting evolving concepts.

Most of traditional data stream classification algorithms, including CVFDT, need to supply input with certain data. However, in real-life applications, streaming data contains inherent uncertainty due to imprecise measurement, missing values, privacy protection (Tsang *et al*., 2011), etc. With uncertainty, the value of each data item is represented not by a single value but by multiple values associated with a robability distribution function (pdf) (Qin *et al*., 2009). It is shown that uncertain data is useful for classification task (Tsang *et al*., 2011).

Many classification algorithms for uncertain static datasets have been proposed, such as Uncertain Decision Tree (UDT) (Tsang *et al*., 2011), uncertain Bayes model (Qin *et al*., 2011) and rule-based classifier (Gao and Wang, 2010). However, few works were devoted to classifying uncertain data streams. In our previous work (Liang *et al*., 2012), we proposed puuCVFDT algorithm for uncertain data streams. However, this algorithm only handled positive and unlabeled samples and build binary classifier. Besides, it exploited uncertainty only in training phase.
In this paper, based on CVFDT algorithm, we proposed uCVFDTc, an uncertain CVFDT tree with Uncertain Naive Bayes (UNB) classifiers at tree leaves. uCVFDTc can exploit uncertainty effectively and efficiently in both the learning and classification phases. Experimental results showed that uCVFDTc had a strong ability to handle concept drift in uncertain data streams; the use of UNB at tree leaves had improved the ability to handle concept drift.

PROBLEM DEFINITION

We write X^u for the set of uncertain attributes, X_i^u for the i-th attribute in X^u and X_{it}^u for the value of X_i^u on the t-th sample st. X^u_{it} is represented by a probability distribution over a domain. If X_i^u is categorical, the domain is defined as $Dom(X_i^u) = \{v_1, \ldots, v_m\}$; the distribution is a probability vector $P = <p_{i1}, \ldots, p_{im}>$ such that $P(X_i^u = v_j) = p_{ij}$ and $\Sigma^m_{i=1} p_{ij} = 1$. If X_i^u is

Corresponding Author: Yang Zhang, College of Mechanical and Electronic Engineering, Northwest A and F University, Yangling, 712100, Shaanxi, People Repbulic of China

numerical, the domain is an interval $[a_{it}, b_{it}]$; the distribution is a pdf $f_{it}(x)$ that $\int_{bit}^{uit} f_{it}(x)dx = 1$.

An uncertain data stream is a sequence of samples $S = <s_1, s_2, \ldots, s_t, \ldots>$, where $s_t = <X^u, y>$, $X^u = \{X_1^u, X_2^u, \ldots, X_d^u\}$, $y \in C = \{C_1, C_2, \ldots, C_{|C|}\}$ represents the class label.

The task of this paper is to build a decision tree model on uncertain data stream S incrementally. Any time, the model can be used to predict the class label of unknown sample $s_t = <X^u, y>$.

ALGORITHM

CVFDT algorithm: CVFDT uses the first samples from the stream to select the root attribute; once the root attribute is chosen, the subsequent samples will be passed down to corresponding leaves and used to choose the test attribute there and so on recursively. Each leaf stores not the samples it observes but sufficient statistics about these samples. To choose the best splitting attribute, a Hoeffding test based on Hoeffding bound theory is run on these statistics, as follows:

Let r be a random variable with range R. If n observations of r have been made, with probability 1-δ, the Hoeffding bound ensures the difference between the true mean and observed mean is less than $\varepsilon = \sqrt{R^2 ln(1/\delta)/2n}$. Let $G(X_i)$ be the heuristic evaluation function used to choose test attributes. For information gain, $R = \log_2 C$. After observing n samples at a leaf, let X_a and X_b be the attribute with highest and second highest observed G and $\Delta G = G(X_a) - G(X_b)$. Given a desired δ, if $\Delta G > \varepsilon$, the Hoeffding bound guarantees that X_a is the best attribute with probability 1-δ.

Handling uncertain data: To learn a decision tree from uncertain data streams, uCVFDTc used the above framework. With uncertainty, however, the value of X_i^u is not a single value but multiple values with a pdf. Hence, an uncertain sample should not be simply put into one child but be split into a set of sub-samples before entering a child. Following (Tsang *et al*., 2011), if X_i^u is categorical, s_t will be split into $m = |Dom(X_i^u)|$ sub-samples $\{s_{t1}, s_{12}, \ldots, s_{tm}\}$; these samples will enter corresponding leaves. If X_i^u numerical, st will be split into m = 2 sub-samples $\{s_{t1}, s_{t2}\}$ and enter the left and the right child. If st is weighted with w_t, s_{ti}, will be weighted with w_{ti} such that $\Sigma^m_{i=1} w_{ti} = w_t$.

Maintaining sufficient statistics: For the sake of limited space and efficiency, each leaf of uCVFDTc only stores sufficient statistics about the pdf values of samples it observes. For growing a tree and adapting the tree to the current concept, at each node, uCVFDTc maintains the same sufficient statistics as our previous work (Liang *et al*. 2012). For categorical attribute, statistic is a number n_{ijk}. For numerical attribute, statistic is parameters $(\mu_{ik}, \sigma^2_{ik}, E_{ik})$ corresponding to a Gaussian pdf $g_{ik}(x)$. Hence, we have:

$$g^u_{ik}(x) = \frac{1}{\sqrt{2\pi\sigma^2_{ik}}} e^{-\frac{(x-\mu_{ik})^2}{2\sigma^2_{ik}}}$$

Choosing the split attribute: uCVFDTc uses uncertain information gain (UIG) as $G(X_i)$. It adopts Hoeffding test as described above to select split attribute. Let S_N be the set of samples observed at node N. Suppose attribute X_i^u is chosen as split attribute and it splits set S_N into m sets of sub-samples $\{S_{i1}, S_{i2}, \ldots, S_{im}\}$. Thus, UIG is calculated below (Qin *et al*., 2009):

$$UIG(SN, Xiu) = uEntropy(SN) - \sum_{i=1}^{m} \frac{PC(S_{ij})}{PC(S_N)} \times uEntropy(S_{ij})$$

where $PC(S) = \Sigma_{\delta,\epsilon,S} w_t$ represents probabilistic cardinality of samples in and uEntropy (S_{ij}) denotes expected information entropy, which can be calculated below (Qin *et al*., 2009):

$$uEntropy(N) = -\sum_{k=1}^{|C|} PC(S, C_k)/PC(S) \times \log(PC(S, C_k)/PC(S))$$

where $PC(S, C_k) = \Sigma_{\delta,\epsilon,S} w_t \times P(C(s_t) = C_k)$ is the probabilistic cardinality of samples with class label C_k in S. Obviously, the calculation of UIG requires only cardinalities: $PC(S_N)$, $PC(S_N, C_k)$, $PC(S_{ij})$ and $PC(S_{i1}, C_k)$.

If X_i^u is discrete, S_N is split into $|Dom(X_i^u)|$ subsets. Using statistics stored at node N, required cardinalities can be estimated: $PC(S_{i1}, C_k) = n_{iik}$, $PC(S_N) = \Sigma^{|C|}_{k=1} \Sigma^{m}_{k=1}$, n_{ijk}.

If X_i^u is continuous, S_N is split into m = 2 subsets. Using statistics stored at node N, required cardinalities can also be estimated:

$$PC(S_{i1}, C_k) = \sum_{ik} \int_{-\infty}^{zi} g^u_{ik}(x)dx, PC(S_{i2}, C_k) = \sum_{ik} - PC(S_{i1}, C_k)$$
$$PC(S_N, C_k) = \sum_{ik} - PC(S_N) = \sum_{k=1}^{|C|} PC(S_{i2}) = \sum_{k=1}^{|C|} PC(S_{i2}, C_k)$$

The expected count for n observations at node N is $PC(S_N)$. Hence, we have Hoeffding bound:

$$\varepsilon \sqrt{R^2 ln(1/\delta)/2PC(S_N)}$$

Building uCVFDTc tree: Similar to CVFDT, in training phase, uCVFDTc keeps its model consistent with a sliding window of samples by growing and removing operations. As more samples arrive, if the concept changes, the split previously passed Hoeffding test will no longer do so. In this case, uCVFDTc begins to grow an alternative tree with the new best attribute at its root. When the alternative tree is more accurate, it replaces the old one. Slide window maintaining, split checking and old tree replacing operations are the same as those of CVFDT. For limited space, they are not shown here.

Algorithm 1: uCVFDTc growing process

UCVFDTcGrow (HT, G, s_t, w_t, δ, τ, n_{min})
Input: HT uncertain decision tree,
G(.) heuristic evaluation function used c
St to choose test attributes.
Wt a new coming uncertain sample,
δ the weight of st,
τ one minus the desired probability of
n_{min} choosing the correct attribute a user-supplied tie threshold the number of samples between checks for growth.
Output: HT uncertain decision tree
Let l the root of HT.
Collect sufficient statistics from (s_t, w_t).
For each tree T in ALT (l)
UCVFDTcGrow (T, G, s_t, w, δ, τ, n_{min}).
IF l is not a leaf, than
Split s_t into a sequence of samples
{s_{t1}, s_{t2},..., s_{tm}}
For each sample s_j in {s_{t1}, s_{t2},..., s_{tm}}
Reweight s_j with w_i.
Let l_i be the branch child for s_i.
UCVFDTcGrow (li, G, s_i, w_i, δ, τ, n_{min})
Else
Let n_1, n_2 be the expected count of samples last seen and current seen at l
If the samples seem so far at l are not all of the same class and n_2-n_1>n_{min}, then
Compute G(X^u_i) on sufficient statistics of leaf l
Let X^u_a and X^u_b be the attribute with highest ans second-highest G.

$$\varepsilon\sqrt{R^2\ln(1/\delta)/2PC(Sl)}$$

$$\text{Let } \Delta\overline{G} = G(X^u_a) - G(X^u_b)$$

$$\text{If } \Delta\overline{G} > \varepsilon \text{ or } \Delta\overline{G} \le \varepsilon < \tau, \text{ then}$$

Replece l by an internal node that splits on X^u_a
For each branch of the split.
Add a new leaf l_j
Initiate the sufficient statistics of lead l_j
Return HT

Algorithm 1 shows the pseudo code for growing uCVFDTc tree. In algorithm, each node collects sufficient statistics from samples it observes (Step 1~2). At internal nodes, an uncertain sample is split into m fractional samples. The sub-trees start new growing processes with these samples (Step 5~16). At leaves, the UIG and Hoeffding bound are calculated on sufficient statistics (Step 14~16). Then the Hoeffding test is run (Step 17~18). If it passes, the best attribute is chosen and the leaf node is split into an internal node (Step 10~22). The alternative trees grow similarly (Step 3~4).

Algorithm 2: uCVFDTc removing process

Input: Uncertain decision tree,
s_t an uncertain sample to be removed,
w_t the weight of
ID corresponding to
Output: uncertain decision tree
Let l the root of HT
If ID of l>ID
Return
Remove sufficient statistic corresponding to sample.
For each tree T in ALT (l)
Forget sample (T, s_t, w_t, ID)
If l is not a leaf, then.
Split si into a sequence of samples
{s_{t1}, s_{t2},...,s_{tm}}
For each sample s_i in {s_{t1}, s_{t2},...,s_{tm}}
Reweight s_i with w_i
Let l_i be the branch child for s_i.
Forget sample (l_i, s_i,w_i,ID)
Return HT

Algorithm 2 lists the pseudo code for removing an outdated sample from the tree. Sufficient statistics are removed from corresponding node. At internal nodes, an uncertain sample is split into M fractional samples. The sub-trees start new removing processes with these samples. The alternative trees remove the outdated sample analogously.

Classifying unknown samples: The classification model for $s_t = \langle X^u, y\rangle$ is a function that maps attribute to a class probability distribution <P (C_1), P (C_2), ..., P ($C_{|C|}$)>.. Here, we recursively define function f (s_t, w_t, N) to calculate such distribution. At each internal node N, st is split into m sub-samples. Thus, the function is defined as:

$$f(s_t, w_t, N) = \sum_{i=1}^{m} f(s_t, w_j, N_j)$$

At each leaf, if majority strategy is adopted, the function is defined as $f(s_t, w_t, l) = w_t\langle P(C_1), P(C_2), ..., P(C_{|C|})\rangle$. Here, $P(C_k) = PC(S_l, C_k)/PC(S_l)$. If UNB strategy is adopted, it is defined as $f(s_t, w_t, l) = w_t\langle P(C_1|X^u), P(C_2|X^u), ..., P(C_{|C|}|X^u)\rangle$, where $P(C_i|X^u)$ is the class posterior probability of C_i. According to Bayes theory and assuming attributes are independent,$P(C_i|X^u)$ can be estimated below:

$$P(C_k|X^u) \propto P(C_k)\prod_{i=1}^{d} P(X^u_{it}|C_k)$$

where, $P(C_k) = P|C(S_l, C_k)/PC(S_l)$. Notice that an uncertain attribute would take many values with a pdf, the class conditional probability $P(X^u_{it}|C_k)$ is evaluated by its expected value. If X_i^u is categorical, $P(X^u_i|C_k)$ can be obtained by:

$$P(X^u_{it} \mid C_k) = \sum_{j=1}^{m} P(X^u_{it} = v_j \mid C_k) P(X^u_{it} = v_j)$$
$$= \sum_{j=1}^{m} p_{ij}\, n_{ijk} / PC(S_l, C_k)$$

If X_i^u is numerical, $P(X_i^u \mid C_k)$ is calculated by:

$$P(X^u_{it} \mid C_k) = \int_{a_{it}}^{b_{it}} f_{it}(x) g^u_{ik}(x) dx$$

The integration of above equation can be evaluated through any numerical methods. However, for uniform uncertainty $f_{it}(x) \sim U(a_{it}, b_{it})$, since $g^u_{ik}(x) \sim N(\mu\sigma^2)$, integration can be estimated by:

$$P(X^u_{it} \mid C_k) = \int_{a_{it}}^{b_{it}} f_{it}(x) g^u_{ik}(x) dx$$
$$= 1/(b_{it} - a_{it}) \int_{a_{it}}^{b_{it}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\alpha-\mu}{2\sigma^2}} dx$$
$$= 1/(b_{it} - a_{it})(G(1/(b_{it}) - G(1/(a_{it}))$$

where $G|(x)$ is the probability cumulative function of $g^u_{ik}(x) \sim N(\mu\sigma^2)$,. Thus, the probability distribution of st is estimated by $f(s_t, w_t, R)$, where $w_t = 1$ denotes the weight for all test samples and R is the root of uCVFDTc tree. Thus, st is predicted to have class label:

$$\text{argmax}(P(C_1), P(C_2),..., P(C_{|C|}))$$
$$= \text{ard max } (f(s_t), 1, R))$$

In the rest of this study, we write uCVFDTcMC and uCVFDTcUNB for uCVFDTc with majority and UNB classification strategy, respectively and use uCVFDTc to denote the both.

EXPERIMENTAL STUDY

In this section, we conduct experiments to evaluate the performance of uCVFDTc algorithm, by comparing with UDT and CVFDT algorithm.

Experiment setup: All algorithms are implemented in Java language and run on a machine with Intel i7 3770 cpu@3.4GHz, 8GB RAM and Win7 (64 bit) OS. Since the unavailability of public uncertain data, this section converts synthetic dataset into uncertain one.

Synthetic datase:. Two synthetic datasets are used (1) Moving hyperplane (Wang *et al*., 2003). It is used to simulate gradual concept drift. In experiment, 1000K samples are generated. Each sample contains d = 10 attributes and K = 2 of which are involved in concept change, (2)SEA (Street and Kim, 2001). It simulates abrupt concept drift by changing parameter θ. In experiment, 500K samples are generated and divided into four sets. Parameters θ in each set are set to 5, 9, 6 and 11.

Simulating uncertainty: We use the method described in (Qin *et al*., 2009) to inject uncertainty into categorical and numerical values. For later one, only uniform uncertainty is injected. We use ω to denote uncertain level of a dataset.

The parameters for CVFDT and uCVFDTc are borrowed from (Hulten *et al*., 2001). For each experiment setting, 5 trials of experiments are conducted and averaged results are reported.

Experimental result:

Ability to handle uncertainty: This group of experiments is conducted on moving hyperplane dataset. To evaluate the ability to tackle categorical uncertainty, numerical attributes are uniformly discretized into 5 bins following the method in (Hulten *et al*., 2001). We set ω = 25%. Every 10000 samples throughout the run, all algorithms are tested on the upcoming 10000 samples. Because UDT is a batch learner, the slide window version UDT-W is implemented by running UDT on a slide window of 100000 samples.

Figure 1 compares the learning curve of uCVFDTc and CVFDT. The curve of uCVFDTcMC is very close to

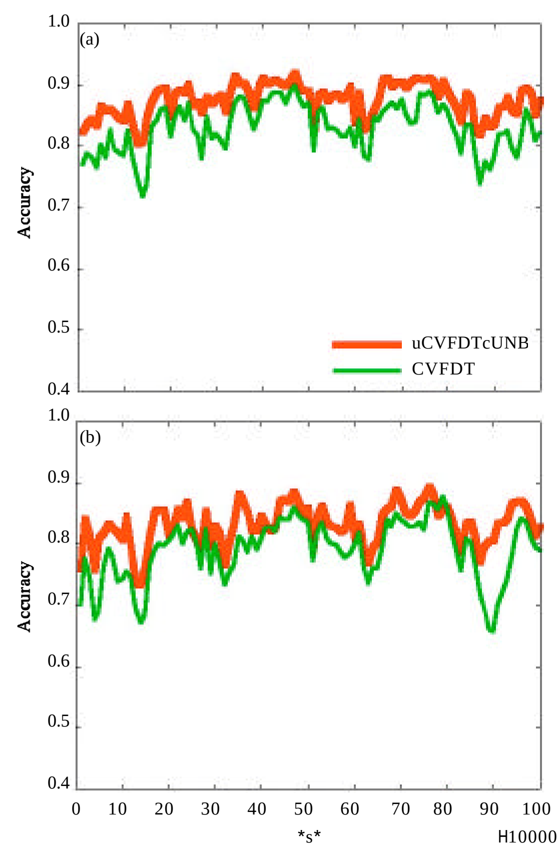

Fig. 1(a-b): Comparing accuracy of uCVFDTc and WVFDT (a) Numerical uncertainty and (b) Categorical uncertainty

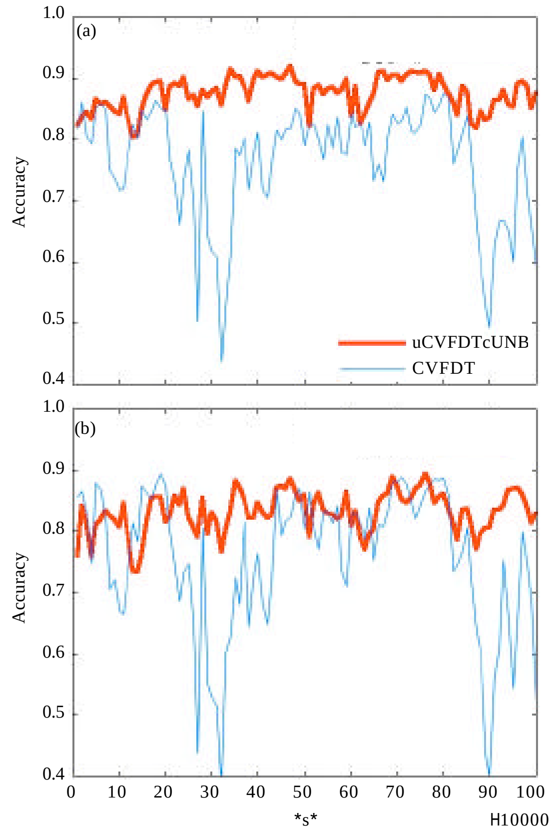

Fig. 2(a-b): Comparing accuracy of uCVFDTc and UDT-W (a) Numerical uncertainty and (b) Categorical uncertainty

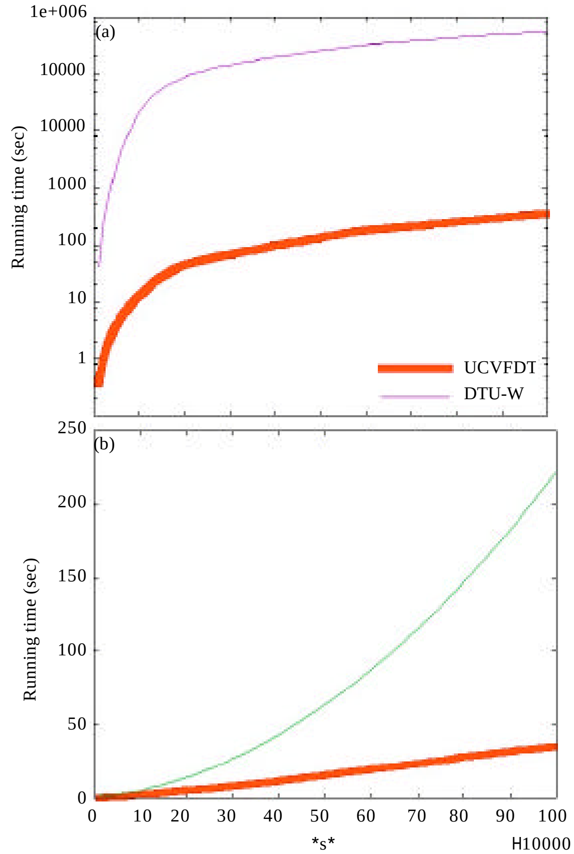

Fig. 3(a-b): Comparing training time of uCVFDTc and CVFDT (a) Numerical uncertainty and (b) Categorical uncertainty

that of CVFDT. For clarity, it is omitted here. It can be observed from Fig. 1 that, the accuracy of uCVFDTcUNB always outperforms that of CVFDT (or uCVFDTcMC). These results show that uCVFDTc can handle the uncertain data with concept drift; the use of UNB can improve the performance at classification stage.

Figure 2 compares the curve of uCVFDTcUNB and UDT-W. Although both uCVFDTcUNB and UDT-W keep their model consistent with the same slide window of 100000 samples, uCVFDTcUNB substantially outperforms UDT-W. This situation suggests that uCVFDTc can reuse the knowledge learned from old uncertain samples and the use of UNB can help to exploit more uncertain information at classification phase and improve the accuracy.

Figure 3 shows the training time needed by uCVFDTc and UDT-W throughout the run. The training time of uCVFDTc does not depend on the classification strategy. It can be observed from Fig. 3 that, throughout the run, uCVFDTc run substantially faster than UDT-W. This is because to learn from uncertain data, uCVFDTc makes only single scan but UDT-W has to make multiple passes. These results suggest uCVFDTc can learn efficiently from uncertain data streams with concept drift.

Ability to handle concept drift: This group of experiments is conducted on both moving hyperplane and SEA datasets. We set. For moving hyperplane, the number of attribute involved in concept drift is set. The test method on this dataset is the same as above experiments. For SEA, every 500 samples throughout the run, all algorithms are tested on the upcoming 1000 samples. Fig. 4 shows the learning curve on moving hyperplane. Similarly, the learning curve of uCVFDTcMC is very close to that of CVFDT. For clarity, it is omitted here. It can be observed from the Fig. 4 that uCVFDTcUNB substantially outperforms CVFDT (or uCVFDTcMC). As concept continuously changes, the accuracy of uCVFDTcUNB performs quite stable and as more attributes involve in concept drift, the differences between uCVFDTcUNB and CVFDT enlarge. This situation shows uCVFDTc has

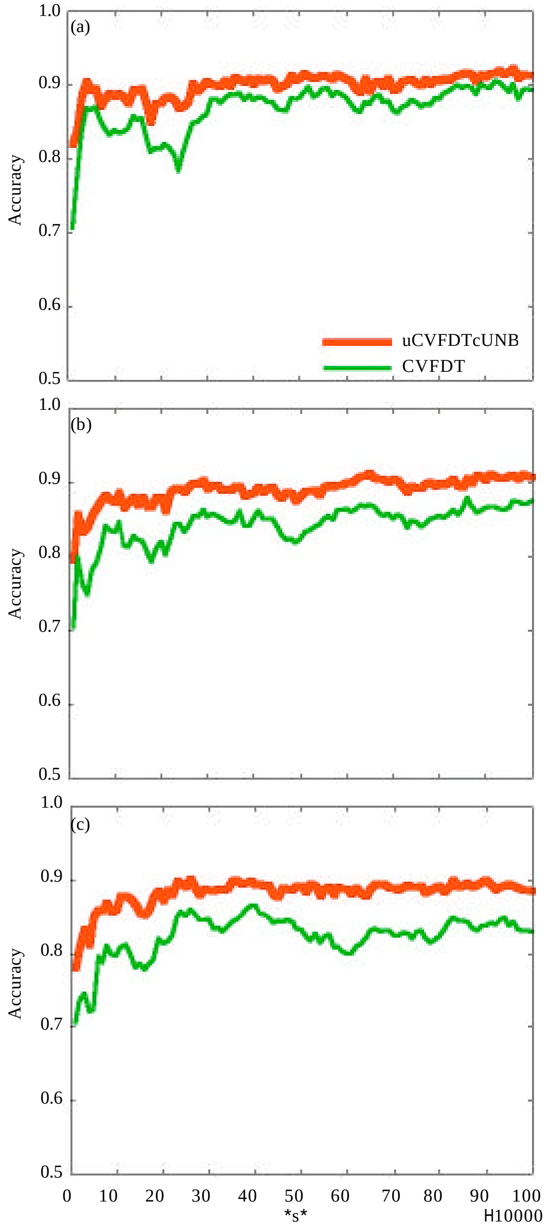

Fig. 4(a-c): Ability to cope with gradual concept drift (a) K = 4, (b) K = 6 and (c) K = 8

strong ability to handle concept drift in uncertain data streams; the use of UNB can improve such ability.

Figure 5 gives experimental result on SEA. There are three drifts on the whole stream. When drift happens, the accuracy of all algorithms suddenly drops and then recovers gradually. For the same reason above, the curve of uCVFDTcMC is omitted here. It can be observed that uVFDTcUNB performs the best recovery. Every time it can recover very fast. These results demonstrate again

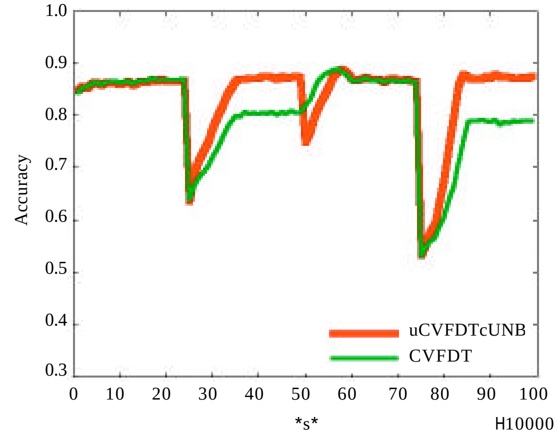

Fig. 5: Ability to handle abrupt concept drift

that uCVFDTc has strong ability to cope with concept drift in uncertain data streams; the use of UNB improve such ability.

CONCLUSIONS

This study proposed a novel algorithm, namely uCVFDTc, to learn decision trees from uncertain data streams. The uCVFDTc algorithm uses Hoeffding bound theory, slide window and alternative tree techniques to build fast decision trees online. At classification stage, it uses UNB to improve accuracy. Experimental results showed that uCVFDTc had strong ability to cope with uncertainty and concept drift; the use of UNB at leaves could improve classification performance, especially the ability to handle concept drift.

ACKNOWLEDGMENT

This study was supported in part by National High-tech R&D Program of China (2013AA10230402).

REFFERENCES

Domingos, P. and G. Hulten, 2000. Mining high-speed data streams. Proceedings of the 6th ACM SIGKDD International Conference on KNOWLEDGE DISCovery and Data Mining, August 20-23, 2000, Boston, Massachusetts, United States, pp: 71-80.

Gama, J., 2012. A survey on learning from data streams: Current and future trends. Prog. Artif. Intell., 1: 45-55.

Gao, C. and J. Wang, 2010. Direct mining of discriminative patterns for classifying uncertain data. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 25-28, 2010, Washington, DC, USA, pp: 861-870.

Hoeffding, W., 1963. Probability inequalities for sums of bounded random variables. J. Am. Stat. Assoc., 58: 13-30.

Hulten, G., L. Spencer and P. Domingos, 2001. Mining time-changing data streams. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 26-29, 2001, ACM Press, New York, pp: 97-106.

Liang, C., Y. Zhang, P. Shi and Z. Hu, 2012. Learning very fast decision tree from uncertain data streams with positive and unlabeled samples. Inform. Sci., 213: 50-67.

Mala, A. and F.R. Dhanaseelan, 2011. Data stream mining algorithms-A review of issues and existing approaches. Int. J. Comput. Sci. Eng., 3: 2726-2732.

Polikar, R., 2006. Ensemble based systems in decision making. IEEE Circuits Syst. Mag., 6: 21-45.

Qin, B., Y. Xia and F. Li, 2009. DTU: A decision tree for uncertain data. Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, April 27-30, 2009, Bankok, Thailand, pp: 4-15.

Qin, B., Y. Xia, S. Wang and X. Du, 2011. A novel bayesian classification for uncertain data. Knowledge Based Syst., 11: 1151-1158.

Quinlan, R.J., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA., USA.

Street, W.N. and Y.S. Kim, 2001. A Streaming Ensemble Algorithm (SEA) for large-scale classification. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 26-29, 2001, San Francisco, CA., USA., pp: 377-382.

Tsang, S., B. Kao, K.Y. Yip, W.S. Ho and S.D. Lee, 2011. Decision trees for uncertain data. IEEE Trans. Knowl. Data Eng., 23: 64-78.

Wang, H., W. Fan, P. Yu and J. Han, 2003. Mining concept-drifting data streams using ensemble classifiers. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 24-27, 2003, Shanghai, China, pp: 226-235.