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 10 (10): 1926-1933, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.1926.1933

A Framework for Classifying Uncertain and Evolving Data Streams

[1]Wenhua Xu, [2]Zheng Qin and [2]Yang Chang
[1]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, Beijing, China
[2]School of Software, Tsinghua University, Beijing 100084, Beijing, China

Abstract: During the last decade, classification from data streams is based on deterministic learning algorithms which learn from precise and complete data. However, a multitude of practical applications only supply approximate measurements. Usually, the estimated errors of the measurements are also available and they are valuable supplemental information for the classification process. Therefore, the development of highly efficient algorithms dealing with uncertain examples has emerged as an exciting new direction in stream data mining literature. In this study, an ensemble classification model ECluds is built from data streams having uncertain attribute values. ECluds applies supervised k-means clustering algorithm on uncertain data stream chunks, then extracts sufficient statistics into micro-clusters. An ensemble of micro-clusters performs classification on test examples using nearest neighbor algorithm and majority voting strategy. Our experiments on synthetic and real-world datasets show that ECluds is highly scalable for data streams and more effective than a purely deterministic method.

Key words: Classification, concept drifts, ensemble classifier, probability density function, standard error, supervised k-means clustering, uncertain data streams

INTRODUCTION

Most available approaches in classifying data streams focused on handling precise and deterministic data examples (Domingos and Hulten, 2000; Wang *et al*., 2003; Bifet *et al*., 2007). However, data streams are inherently accompanied with uncertainty due to a multitude of reasons (Guang *et al*., 2009). The characteristic of uncertainty in data streams should be considered. Otherwise, the induced model could be unreliable. Therefore, the development of highly efficient algorithms dealing with uncertain examples is a new direction in stream data mining literature.

The uncertainty is basically referred to as the level of imprecise which can be quantified in some way (Aggarwal, 2009). It can exist in attribute values as well as in class labels of examples. In this study, the classification of data streams whose attribute values are uncertain is explored.

The uncertainty of attribute values is usually specified in the form of Probability Density Functions (PDF) (Pan *et al*., 2010; Qin *et al*., 2009a). However, the entire PDF of the examples are unavailable sometimes in real applications. Therefore, a less restrictive condition is adopted which assumes that the standard error of examples is known. This is a more practical and flexible approach to handle uncertainty information.

An ensemble model ECluds is developed based on the clustering algorithm to classify uncertain data streams with concept drift. ECluds applies supervised k-means clustering algorithm on uncertain data stream chunks, then extracts sufficient statistics from clusters into micro-clusters. An ensemble of micro-clusters performs classification on test examples using nearest neighbor algorithm and majority voting strategy. Evaluations on both synthetic and real-world datasets show that ECluds maintains highly competitive accuracy, speed and scalability and more effective than purely deterministic approaches.

Stream data classification: The problem of stream data classification is that: Given an infinite amount of continuous data examples, how to model them in order to capture their trends and patterns and make time-critical predictions (Wang *et al*., 2003). The classification algorithms are facing two challenges: the infinite volume of stream data and the concept drift (Masud *et al*., 2008). In other words, the concept driven from new examples is constantly evolving, so that the new concept must be incorporated into the model without repeating the entire learning process.

Two strategies can be adopted: single model classification and ensemble model classification. Single model classification techniques incrementally update the

Corresponding Author: Wenhua Xu, Department of Computer Science and Technology, Tsinghua University, 100084, Beijing, China Tel: 0086-10-51537927

models with new data during the training. Very Fast Decision Tree (VFDT) is one of the most successful and prominent algorithms specifically designed for classifying deterministic data streams (Domingos and Hulten, 2000). A number of extensions of VFDT have been proposed to enhance its performance or adaptability, such as Concept-adapting Very Fast Decision Tree (CVFDT) (Hulten *et al*., 2001), VFDTc (Gama *et al*., 2003), Hoeffding Option Tree (Pfahringer *et al*., 2007), Adaptive Hoeffding Option Tree (Bifet *et al*., 2009) Hoeffding Perceptron Tree (Bifet *et al*., 2010), etc.

Single model techniques are usually difficult to cope with concept drift. To overcome the weakness, some ensemble techniques have been proposed (Wang *et al*., 2003; Scholz and Klinkenberg, 2005; Masud *et al*., 2008; Bifet *et al*., 2009). These ensemble approaches have the advantage that they can be more efficiently built than updating a single model and they usually achieve higher accuracy.

Uncertain data classification: Some previous work focused on building classification models on uncertain data examples. All of them assume that the PDF of attribute values are known. Bi and Zhang (2004) presented a general statistical framework to build a Support Vector Machine (SVM) model on input data corrupted with uncertainty. Qin *et al*. (2009a) proposed a rule-based model eRule for classifying uncertain data. Tsang *et al*. (2009) developed an Uncertain Decision Tree (UDT) for uncertain data. Qin *et al*. (2009b) developed another type of decision tree DTU for uncertain data classification. Ge *et al*. (2010) proposed a neural network method for classifying uncertain data.

Uncertain stream data classification: Since the uncertainty is prevalent in data streams, the research of classification is dedicated to uncertain data streams nowadays. Unfortunately, only a few algorithms are available. Pan *et al*. (2010) proposed two types of ensemble classification algorithms, Static Classifier Ensemble (SCE) and Dynamic Classifier Ensemble (DCE), for mining uncertain data streams. Liang *et al*. (2010) proposed a CVFDT based decision tree named UCVFDT for uncertain data streams. UCVFDT has the ability to handle examples with uncertain attribute values by adopting the model described by Qin *et al*. (2009a) to represent uncertain nominal attribute values.

Besides above methods, there is also research on clustering uncertain data streams with concept drift. Aggarwal and Yu (2008) proposed a framework for clustering uncertain data streams. They designed uncertain micro-clusters to track the statistics of the stream and leveraged them for the clustering process.

ENSEMBLE CLASSIFIER FOR UNCERTAIN DATA STREAMS

The classification problem for data streams is generally defined as follows. A set of infinite and evolving training examples $S = \{(x_t, y_t) | t = 1,..., N,...\}$ of the form (x, y) is given, where $x_t = (x_{t1},...,x_{tD})$ is a value vector of D-dimensional attributes, each of which may be numerical or nominal. Y_t is a nominal class label, that is $y_t \in \{class_1,....,class_j\}$. The stream data classification is to build a model y = f(x) from these examples that will predict the class labels of future examples with high accuracy.

The uncertain data models: When a numerical attribute value x_{id} is uncertain, it is referred to as an Uncertain Numerical Attribute (UNA). In this study, the value of UNA is treated as a continuous random variable and denoted by the standard error model introduced by Aggarwal and Yu (2008).

The uncertainty of the standard error model is characterized by the attribute value x_{id} and the associated estimated error $e_{id}(x)$. The exact function $e_{id}(x)$ is usually unknown, yet the standard deviation of the error can be measured. Therefore, it is adopted to represent the uncertainty information and is denoted by $\Psi_{td}(x) = SD[e_{id}(x)]$. The mean of the error is assumed to be 0, that is $E[e_{id}(x)] = 0$. The corresponding D-dimensional error vector of the example x_t is denoted by $\Psi_{td}(x) = (\Psi_{t1}(x) \Psi_{tD}(x))$. Therefore, the tth training example is denoted by the triple $(x_t, \Psi_t(x), y_t)$ and the classification model is denoted by $y = f(x, \Psi(x))$.

The ECluds algorithm overview: The ECluds algorithm (the acronym for Ensemble Classifier for uncertain data streams) will be introduced in this subsection. Its training and classification process follows from the ensemble deterministic algorithm SmSCluster (Masud *et al*., 2008) and is shown in Fig. 1. The training examples are divided into chunks of same size $S = \{S_1,....S_n,...\}$ and:

$$\begin{aligned} S_1 &= \{(x_1, y_1),...,(x_M, y_M)\} \\ &\cdots\cdots \\ S_n &= \{(x_{(n-1)M+1}, y_{(n-1)M+1}),...,(x_{nM}, y_{nM})\} \end{aligned} \quad (1)$$

where, M is the chunk size, S_n is the nth data chunk. The following three operations are performed step by step iteratively.

Training: Each model E_n is trained by applying supervised uncertain k-means clustering algorithm on a training chunk S_n. After k-clusters are generated, the sufficient statistics information is extracted from them and stored as micro-clusters. After that, the original examples are discarded to release memory. The set of micro-clusters is the base classification model E_n.

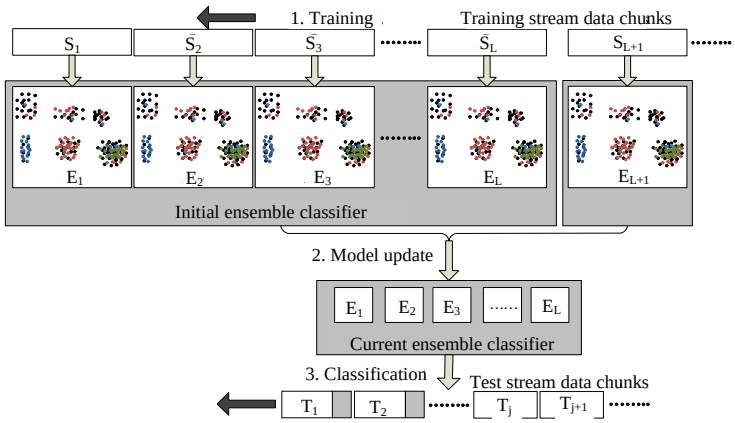

Fig. 1: Training and classification process of ECluds algorithm

Model update: The initial ensemble model E consists of L base models, i.e., E = {E_1,...E_L} which are induced from the first L training chunks. When a new chunk S_{L+1} arrives, a new model is built from it. Then L models are selected from the L + 1 models based on their accuracies on chunk S_{L+1}. The worst classifier is removed and a new one is added to the ensemble. E always keeps L best base models. This step assures that the fresh knowledge can be incorporated into the ensemble model and the outdated concept be discarded in order to adapt concept drift.

Classification: Any time to classify a test example in the test dataset T, we apply the nearest neighbor algorithm and find the nearest micro-cluster from each base model in E. Then we select the major label in the L clusters. This label is the predicted label for the test example.

The supervised k-means clustering: The supervised uncertain k-means cluster algorithm is applied to build a base classifier when a training data chunk S_n is ready. It is an extension of the conventional unsupervised k-means clustering algorithm.

Given a data chunk S_n and an integer constant k, the problem of unsupervised k-means clustering (Velmurugan and Santhanam, 2011; Vijendra, 2011) is to assign examples to k clusters {C_1,...,C_k} whose centroids are {c_1,...,c_k}, so that the following objective function is minimized based on a distance metric function:

$$O_{kmeans} = \sum_{i=1}^{k} \sum_{x_t \in C_i} d(x_t, c_i). \quad (2)$$

Different clustering algorithms apply different distance metric functions and different objective functions. The distance metric function d (x_t, c_i) used in this study is the square of Euclidean norm between an example x_t and a cluster centroid c_i which can be formulated as:

$$d(x_t, c_i) = \|x_t - c_i\|^2 = (x_t - c_i) \cdot (x_t - c_i) \quad (3)$$

The goal for supervised k-means clustering is to minimize the intra-cluster dispersion and meanwhile minimize the impurity of each cluster (Garg and Jain, 2006). A cluster is completely pure if it contains examples all from the same class. Therefore, the objective function for supervised k-means clustering is as follows:

$$O_{s-kmeans} = \sum_{i=1}^{k} \sum_{x_t \in C_i} d(x_t, c_i) + \sum_{i=1}^{k} w_i \cdot imp_i \quad (4)$$

where, w_i is the weight associated with cluster c_i and imp_i is the impurity metric for C_i. w_i is formulated as follows (Masud *et al*., 2008):

$$w_i = \sum_{x_t \in C_i} d(x_t, c_i) \quad (5)$$

Therefore,

$$O_{s-kmeans} = \sum_{i=1}^{k} \sum_{x_t \in C_i} d(x_t, c_i) \cdot (1 + imp_i) \quad (6)$$

The supervised k-means clustering for uncertain data: In the case of uncertainty, the computations of cluster centroids and distance metric are significantly changed by the probabilistic nature of examples. Therefore, supervised uncertain k-means clustering algorithm is

developed to incorporate the uncertainty into the clustering process. The deterministic distance metric function d (x_t, c_i) is substituted by the expected distance ED (x_t, c_i). Therefore, the objective function for supervised uncertain k-means clustering is as follows:

$$O_{su-kmeans} = \sum_{i=1}^{k} \sum_{x_t \in C_i} ED(x_t, c_i) \cdot (1 + imp_i). \quad (7)$$

For the standard error model, the centroid of a cluster is treated as a random variable vector and computed as the mean values and the mean error terms of the uncertain examples in the cluster. Therefore, the centroid is formulated as:

$$c_i = \frac{1}{|C_i|}\left[\sum_{x_s \in C_i} x_s + \sum_{x_s \in C_i} e(x_s)\right] \quad (8)$$

$$c_{id} = \frac{1}{|C_i|}\left[\sum_{x_s \in C_i} x_{sd} + \sum_{x_s \in C_i} e_{sd}(x)\right] \quad (9)$$

where, $|C_i|$ is the number of examples in cluster C_i.

Given E[e_d (x)] the following equation holds:

$$E(\|c_i\|^2) = \frac{1}{|C_i|^2}\sum_{d=1}^{D}\left[\left(\sum_{x_s \in C_i} x_{sd}\right)^2 + \sum_{x_s \in C_i} \psi_{sd}^2(x)\right] \quad (10)$$

And the expected distance between an example x_t and a centroid c_t is formulated as:

$$\begin{aligned} ED(x_t, c_i) &= E(\|x_t - c_i\|^2) \\ &= E(\|x_t\|^2) + E(\|c_i\|^2) - 2E(x_t) \cdot E(c_i) \end{aligned} \quad (11)$$

where,

$$E(x_t) \cdot E(c_i) = \frac{1}{|C_i|}\sum_{d=1}^{D}\left(x_{td} \cdot \sum_{x_s \in C_i} x_{sd}\right) \quad (12)$$

$$\begin{aligned} E(\|x_t\|^2) &= \sum_{d=1}^{D}\left[x_{td}^2 + \psi_{td}^2(x)\right] \\ &= \|x_t\|^2 + \|\psi_t(x)\|^2 \end{aligned} \quad (13)$$

Therefore, by substitute the Eq. 10, 12 and 13 to Eq. 11, ED (x_t,c_i) is formulated as:

$$\begin{aligned} ED(x_t, c_i) = \|x_t\|^2 + \|\psi_t(x)\|^2 + \frac{1}{|C_i|^2}\sum_{d=1}^{D}\left[\left(\sum_{x_s \in C_i} x_{sd}\right)^2 + \sum_{x_s \in C_i} \psi_{sd}^2(x)\right] \\ - \frac{2}{|C_i|}\sum_{d=1}^{D}\left(x_{td} \cdot \sum_{x_s \in C_i} x_{sd}\right) \end{aligned} \quad (14)$$

Solve the supervised uncertain k-means clustering: The goal of supervised uncertain k-means clustering algorithm is to minimize the objective function Eq. 7. It can be achieved by applying Expectation Maximum (EM) algorithm and perform E-step and M-step iteratively until the convergence condition is fulfilled. The complete algorithm is presented in Algorithm 1. Annotations are added to the pseudo code of the algorithm.

In E-step, we assign each example x_t to a cluster C_i such that the following objective function is minimized:

$$O_{su-kmeans}(x_t) = ED(x_t, c_i) \cdot (1 + imp_i) \quad (15)$$

In M-step, the cluster centroids are recalculated according to the current examples they have.

Algorithm 1: Supervised uncertain k-means clustering

input: {(x_t, Ψ_t (x), y_t|t = 1,...M)} // a set of uncertain examples
output: k-clusters
procedure su-kmeans:

1 Make initial guesses for the cluster centroid c_1,...c_k
2 repeat
3 for t = 1 to M // E-step
4 Assign x_t to cluster C_i where Eq. 15 is minimized
5 end for
6 for I = 1 to k // M-step
7 Recalculate cluster centroid c_i
8 end for
9 until no example changes its cluster // convergence
10 return k clusters
end procedure

Prediction with ECluds: When the clustering process is finished, the sufficient statistics information of clusters are extracted and stored as micro-clusters. They will act as a base classification model. When classifying a test example, two steps are performed. First, find the nearest micro-cluster in each base model by calculating the expected distance between the test example and the centroids of the micro-clusters. Second, select the class with the highest votes from the L clusters as the predicted class label.

EXPERIMENTAL EVALUATION

ECluds was applied on both synthetic datasets and real-world datasets to evaluate its performance in three aspects which were accuracy, running time and scalability, sensitivity to parameters. As currently there is no other methods can handle the standard error uncertainty, the deterministic baseline method SmSCluster was used to compare against ECluds.

Datasets and experiment setup: There is a shortage of publicly available large real-world datasets that are suitable for the evaluation of data stream methods. Thus we used several synthetic datasets that can be found in the literature. They were moving Hyperplane (HP),

Random Radial Basis Function (RRBF) and SEA. Artificial concept drift was introduced to them in order to simulate the real-world evolving nature.

HP was used to compare CVFDT with VFDT under the environment of concept drift by Hulten *et al.* (2001). A hyperplane in D-dimensional space is a set of points x_i which satisfy the equation:

$$\sum_{d=1}^{D} w_d x_{id} = w_0$$

where, w_d is the dimensional weight. Examples satisfying

$$\sum_{d=1}^{D} w_d x_{id} \geq w_0$$

are labeled positive, others are labeled negative. Concept drift is introduced by tuning w_i after each example is generated.

RRBF generates examples from a fixed number of random centroids. Each time a centroid is randomly selected and a data point around the centroid is randomly determined. The centroid gives the class label and the data point determines the attribute values. Concept drift is introduced by moving the centroids with constant speed.

SEA contains abrupt concept drift. It generates examples with three attributes and two class labels (Street and Kim, 2001). N random examples x_t in a 3-dimensional space are generated with each value in [0,10]. After that, the examples are divided into 4 equal blocks to denote different concepts. In each block, an example is labeled positive or negative by using inequality $x_{t1} + x_{t2} \leq \theta$, where, θ is usually 8, 9, 7 and 9.5 for each of the 4 blocks.

The real-world dataset is Forest Covertype (FC) which was obtained from the UCI repository. It contains 581,012 examples, 54 attributes including 10 numeric attributes, 44 binary attributes and 7 class labels. All the 10 numeric attributes were used in our experiments.

We also introduced synthetic uncertainty information into the datasets. The generative approach for the standard error model follows from Aggarwal and Yu (2008). An error is added to a deterministic attribute value x_{id} to generate uncertainty. Letting σ_d^0 be the standard deviation of the entire dataset along the dimension d, σ_d is defined as a uniform random variable drawn from the interval $[0, \eta \cdot \sigma_d^0]$, where η is the error level and usually $\eta \leq 6$. Then, for the dimension d, error which is drawn from the Gaussian distribution with zero mean and standard deviation σ_d is added. $\eta 0$ denotes a deterministic dataset.

MOA (Massive Online Analysis) (Bifet *et al.*, 2007) is a software environment for stream data mining, including classification, clustering and performance evaluations. The uncertain stream data classification algorithm ECluds and the baseline deterministic algorithm SmSCluster were implemented in MOA. The experiments were performed on a 2.0 GHz Intel Core Duo PC with 2 GB RAM, running Windows XP.

Classification accuracy: Experiments were conducted to evaluate ECluds and SmSCluster using both synthetic datasets and real-world dataset with different error levels. The evaluation strategy was interleaved test-then-train: Each example is used for testing the model before using it to train. It assures that the model is always being tested on examples it has not seen, whilst all examples can be used in both training and test.

The information of each dataset is listed in Table 1. Parameter D is the total number of attributes in a dataset, D_c is the number of attributes which are involved in concept drift and D_e is the number of attributes which are added errors. Parameters are set to the default values which are k = 50 (number of clusters), ChunkSize = 5000 (number of examples in each chunk) and L = 8 (ensemble size) for both SmSCluster and ECluds.

Figure 2 illustrates how the average classification accuracy varies with the increasing error level. The X-axis

Table 1: Properties of the datasets

Dataset	Classes	D	D_c	D_e
HP	2	20	4	8
RRBF	4	10	10	4
SEA	2	3	2	2
FC	7	10	-	4

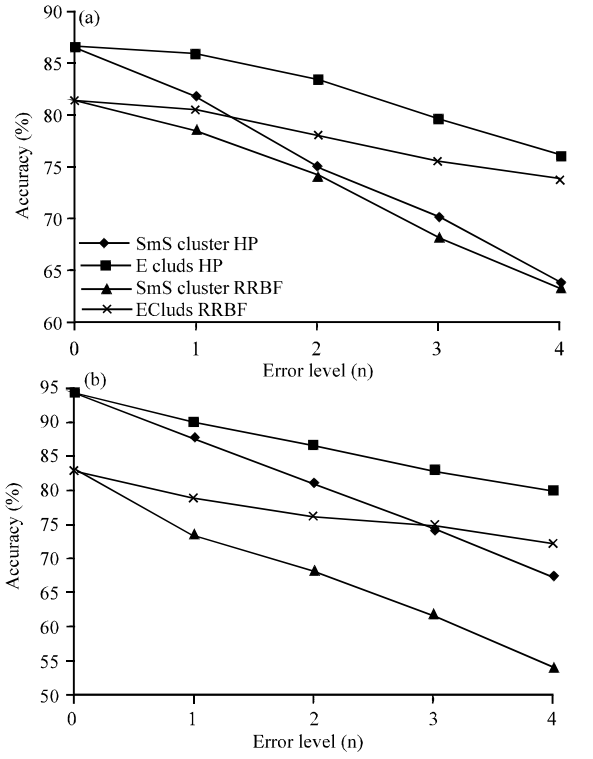

Fig. 2(a-b): Accuracy with increasing error level using two methods

represents the increasing error level and the Y-axis represents the evaluation accuracy.

It can be observed that both of the methods achieve the best accuracy when error level is 0. The reason is evident. ECluds consistently achieves better accuracies than SmSCluster when error level increases. This is because that by exploiting the uncertainty information, an example can be assigned to an appropriate cluster, making the boundary of examples from different classes more accurate. The results indicate that the uncertainty information does help promote the learning effectiveness of ECluds. The results also show that the accuracies of SmSCluster and ECluds decrease with increasing error level. However, ECluds does not decrease significantly.

Figure 3 and 4 show the learning curves of two algorithms on HP η2 dataset and FC η2 dataset. The X-axis represents the number of examples that have been learned and the Y-axis represents the evaluation accuracy. It can be learned from these figures that by exploiting the available uncertain information, the accuracy of ECluds is higher than SmSCluster. Figure 3 shows that though HP dataset contains gradual concept drift, by applying the ensemble learning framework, the accuracy can still be improved constantly with more examples being learned. For the real-world FC dataset, it contains much noise and the precise concept drift is not known. Therefore, Fig. 4 shows that the learning curves fluctuate more frequently than HP dataset.

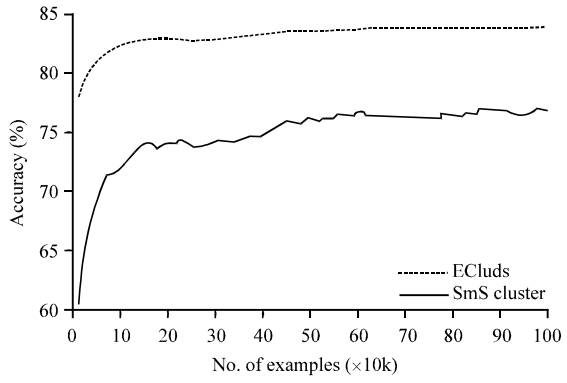

Fig. 3: Learning curves on HP dataset with error level 2

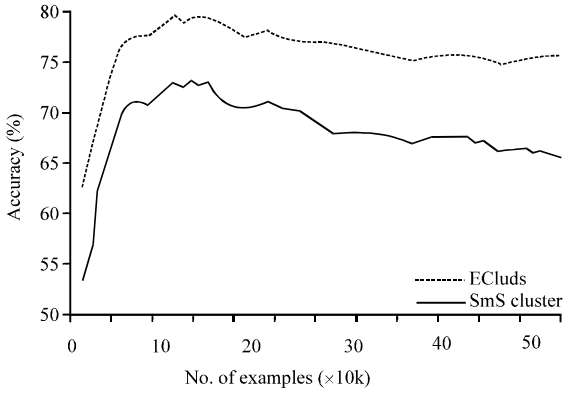

Fig. 4: Learning curves on FC dataset with error level 2

Runtime and scalability: Figure 5 and 6 report the scalability of ECluds on high-dimensional and multi-class data. The training and test time on RRBF dataset comprising 100,000 examples are measured. Figure 5 illustrates how the runtime varies with the number of attributes. It can be noted that the runtime of both methods increases linearly with the number of attributes. SmSCluster is faster since it works with the attribute values only. ECluds needs to devote more time to dealing with the uncertainty information, since each attribute of an example is described by two values at least, resulting double sized input. Moreover, the requirement for computing expected distance is also much greater. It can

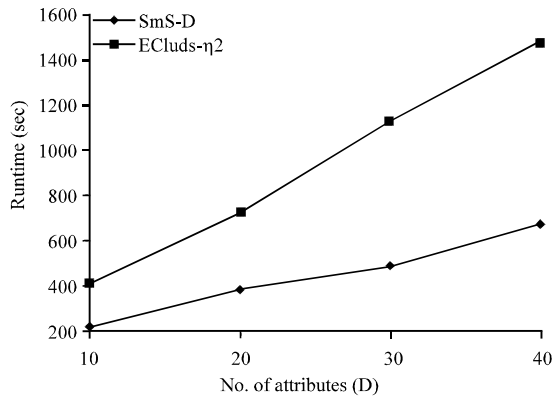

Fig. 5: Runtime curves on RRBF dataset varies with number of attributes

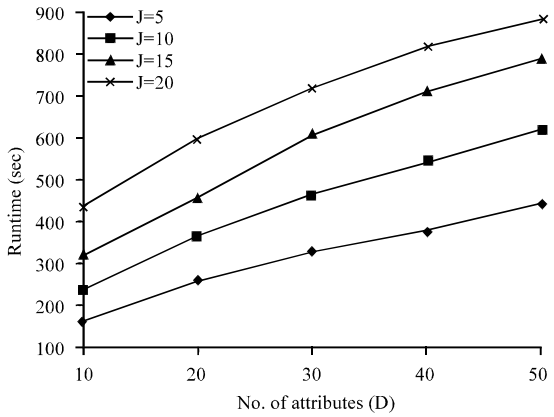

Fig. 6: Runtime curves on RRBF dataset, that varies with number of clusters

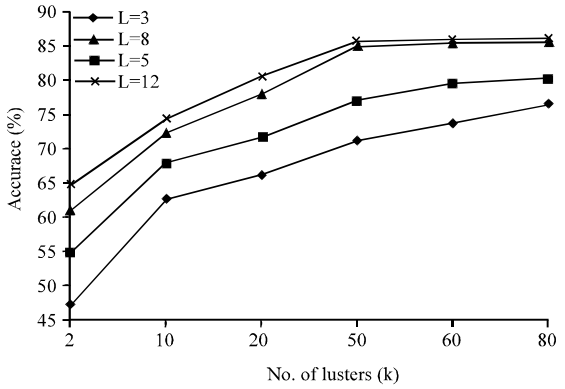

Fig. 7: Sensitivity to parameters k and L on HP dataset with error level 2

be observed from Fig. 5 that ECluds spends about 80-130% more time than SmSCluster to process the same sized dataset.

It can also be observed that the runtime increases linearly with the number of clusters (k) and the number of classes (J) from Fig. 6. This is because that the runtime of computing expected distance and classifying unlabeled examples are all proportional to the number of clusters and classes. It is a desirable characteristic for ECluds since it can scale linearly to higher dimensionality, clusters and class labels.

Sensitivity to parameters: Figure 7 shows how the classification accuracy varies for ECluds with number of clusters and ensemble size. The result is obtained from the HP $\eta 2$ dataset comprising 1,000,000 examples. We observe that higher values of k lead to higher accuracies. The reason is that when k is larger clusters are getting smaller and purer. Therefore, a test example is more similar with its nearest neighbor, leading to a more accurate base model. The accuracy is close to the optimum after k reaches 50. We can also see that the accuracy improves with increasing ensemble size. It has been proved that the average variance of the ensemble model can be reduced with more base models (Wang *et al*., 2003). However, when the ensemble size increases, more out-dated knowledge will be kept in the ensemble model, making it insensitive to concept drift. Therefore, the accuracy may not be improved any more.

CONCLUSIONS

In this study, we address the issue of uncertain stream data classification. The ensemble model ECluds is developed to learn from data streams having uncertain attribute values. ECluds applies supervised k-means clustering algorithm on uncertain data stream chunks, then extracts sufficient statistics into micro-clusters. An ensemble of micro-clusters performs classification on test examples using nearest neighbor algorithm and majority voting strategy. Experiments on synthetic and real-world datasets show that ECluds is highly scalable for data streams and more effective than a purely deterministic method.

Mining from data streams having uncertain class labels is another important issue. Algorithms have been proposed to solve the problem. In future, we would like to develop an uncertain model to describe such scenarios and incorporate the model into the ECluds framework, making ECluds more generic.

ACKNOWLEDGMENT

This study is supported by the National Natural Science Foundation of China (No. 60673024) and the "Eleventh Five" Preliminary Research Project of PLA (No. 102060206).

REFERENCES

Aggarwal, C.C. and P.S. Yu, 2008. A framework for clustering uncertain data streams. Proceedings of the 24th International Conference on Data Engineering, April 7-12, Cancun, pp: 150-159.

Aggarwal, C.C., 2009. Managing and Mining Uncertain Data. Springer Publishing Company, New York.

Bi, J. And T. Zhang, 2004. Support vector classification with input data uncertainty. Adv. Neural Inform. Proc. Syst., 16: 161-168.

Bifet, A., R. Kirkby, G. Holmes and B. Pfahringer, 2007. MOA: Massive Online Analysis. http://sourceforge.net/projects/moa-datastream

Bifet, A., G. Holmes, B. Pfahringer, R. Kirkby and R. Gavalda, 2009. New ensemble methods for evolving data streams. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, June 28-July 1, Paris, France, pp: 139-148.

Bifet, A., G. Holmes, B. Pfahringer and E. Frank, 2010. Fast perceptron decision tree learning from evolving data streams. Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining, June 21-24, Hyderabad, India, pp: 299-310.

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, Aug. 20-23, Boston, Massachusetts, United States, pp: 71-80.

Gama, J., R. Rocha and P. Medas, 2003. Accurate decision trees for mining high-speed data streams. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Oct. 28-31, ACM Press, pp: 523-528.

Garg, S. and R.C. Jain, 2006. Variations of K-mean algorithm: A study for high-dimensional large data sets. Inform. Technol. J., 5: 1132-1135.

Ge, J., Y. Xia and C.H. Nadungodage, 2010. A neural network for uncertain data classification. Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining, June 21-24, Hyderabad, India, pp: 449-460.

Guang, L., W. Ya-Dong and S. Xiao-Hong, 2009. A privacy preserving neural network learning algorithm for horizontally partitioned databases. Inform. Technol. J., 9: 1-10.

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, Aug. 26-29, ACM Press, New York, pp: 97-106.

Liang, C., Y. Zhang and Q. Song, 2010. Decision tree for dynamic and uncertain data streams. JMLR: Workshop Conf. Proc., 13: 209-224.

Masud, M.M., J. Gao, L. Khan, J. Han and B. Thuraisingham, 2008. A practical approach to classify evolving data streams: Training with limited amount of labeled data. Proceedings of the 8th International Conference on Data Mining, Dec. 15-19, Pisa, Italy, pp: 929-934.

Pan, S., K. Wu, Y. Zhang and X. Li, 2010. Classifier ensemble for uncertain data stream classification. Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining, May 24-27, Shenzhen, China, pp: 488-495.

Pfahringer, B., G. Holmes and R. Kirkby, 2007. New options for hoeffding trees. Adv. Artif. Intell., 4830: 90-99.

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

Qin, B., Y. Xia, S. Prabhakar and Y. Tu, 2009b. A rule-based classification algorithm for uncertain data. Proceedings of the 25th IEEE International Conference of Data Engineering, March 29-April 2, Shanghai, China, pp: 1633-1640.

Scholz, M. and R. Klinkenberg, 2005. An ensemble classifier for drifting concepts. Proceedings of the 2nd International Workshop on Knowledge Discovery in Data Streams, (KDDS'05), Porto, Portugal, pp: 53-64.

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, Aug. 26-29, San Francisco, CA., USA., pp: 377-382.

Tsang, S., B. Kao, K.Y. Yip, W. Ho and S.D. Lee, 2009. Decision trees for uncertain data. Proceedings of the 25th IEEE International Conference of Data Engineering, March 29-April 2, Shanghai, China, pp: 441-444.

Velmurugan, T. and T. Santhanam, 2011. A survey of partition based clustering algorithms in data mining: An experimental approach. Inform. Technol. J., 10: 478-484.

Vijendra, S., 2011. Efficient clustering for high dimensional data: Subspace based clustering and density based clustering. Inform. Technol. J., 10: 1092-1105.

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, Aug. 24-27, Shanghai, China, pp: 226-235.