

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224581406>

# An incremental decision tree learning methodology regarding attributes in medical data mining

Conference Paper · August 2009

DOI: 10.1109/ICMLC.2009.5212333 · Source: IEEE Xplore

---

CITATIONS

35

---

READS

228

2 authors, including:



[Derek F. Wong](#)

University of Macau

126 PUBLICATIONS 510 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Computer Security [View project](#)



Chit-Chat [View project](#)

# AN INCREMENTAL DECISION TREE LEARNING METHODOLOGY REGARDING ATTRIBUTES IN MEDICAL DATA MINING

SAM CHAO, FAI WONG

Faculty of Science and Technology, University of Macau, Taipa, Macau  
E-MAIL: lidiase@umac.mo, derekfw@umac.mo

## Abstract:

Decision tree is one kind of inductive learning algorithms that offers an efficient and practical method for generalizing classification rules from previous concrete cases that already solved by domain experts. It is considered attractive for many real-life applications, mostly due to its interpretability. Recently, many researches have been reported to endow decision trees with incremental learning ability, which is able to address the learning task with a stream of training instances. However, there are few literatures discussing the algorithms with incremental learning ability regarding the new attributes. In this paper, *i*<sup>2</sup>Learning (Intelligent, Incremental and Interactive Learning) theory is proposed to complement the traditional incremental decision tree learning algorithms by concerning new available attributes in addition to the new incoming instances. The experimental results reveal that *i*<sup>2</sup>Learning method offers the promise of making decision trees a more powerful, flexible, accurate and valuable paradigm, especially in medical data mining community.

## Keywords:

Incremental learning; learning regarding attributes; decision tree; medical data mining

## 1. Introduction

Inductive learning is a well-known data mining approach to acquire knowledge automatically [1], where decision tree is one kind of inductive learning algorithms that offers an efficient and practical method for generalizing classification rules from previous concrete cases that already solved by domain experts, while its goal is to discover knowledge that not only has a high predictive accuracy but is also comprehensible to users. Decision tree inductive inference is considered attractive for many real-life applications, mostly due to its interpretability [2]. Such technology is well suited for medical diagnosis, whereas the automatically generated diagnostic rules slightly outperformed the diagnostic accuracy of physician specialists [3].

However, the conventional batch learning methods

discard the existing rules and regenerate a set of new rules from scratch once a new instance is introduced. Such reproduction wastes the previous knowledge and effort while its cost may be too expensive. Therefore, incremental learning concept is raised to compensate for the limitations of batch mode algorithms, which is able to build and refine a concept in a step-by-step basis as a new training case is added. Such learning strategy may be more economical than rebuilding a new one from scratch, especially in real-time applications. In which, ID5 [4] and ID5R [5] were the pioneers of incremental learning systems. ID5R is a successor of ID5 algorithm, both of them adopt a pull-up strategy for the tree restructuring process that recursively promotes a new best attribute to the root of the tree or sub-tree and then demotes the original one. The only exception where ID5 is not identical to ID5R is that after pull-up a new best attribute to the root, the sub-trees are not restructured recursively. Afterwards, ID5R has been superseded by ITI (Incremental Tree Induction) [6], which is an enhanced algorithm that is able to handle numeric attributes and missing values, and uses the information gain ratio measure as an attribute selection policy [7]. Nevertheless, all these incremental learning algorithms only address the problems caused by introducing the new training instances after a tree being trained; whereas when a new attribute is provided, they are unable to deal with the new data without relearning again the entire decision tree.

In the next section, the motivation of this paper is described by giving the real medical cases. Then in the third section, a novel incremental learning algorithm with the ability to handle the new available attributes as well as new training instances is proposed and introduced in detail. The evaluation of the proposed method on some real-life datasets is performed in section four. Finally, the limitations of the method and the directions for our further research are discussed and concluded to end this paper.

## 2. Motivation

As we know, learning in real world is interactive, incremental and dynamical in multiple dimensions, where new data could be appeared at anytime from anywhere and of any type. A valuable data mining learning process should be mimicked thoroughly such style to become humanoid and intelligent. In reality, the learning environment is prone to dynamic instead of static. Especially in medical domain, the new viruses or diseases may emerge on the go, while the new symptoms and/or new medical cases must be taken into consideration and added to the current learning model accordingly.

For instance, when diagnosing a general arrhythmia disease, the diagnostic conclusion may be determined according to a patient's corresponding symptoms, medical history, physician's auscultation and a routine ECG (Electrocardiogram) test. Yet regular ECG test is incapable to discover the unusual type of arrhythmia like intermittent arrhythmia, while a more advanced DCG (Dynamic Cardiogram) test should be considered and taken place. Besides, since the outbreak of lethal SARS (Severe Acute Respiratory Syndrome) in year 2003, thereafter once a patient got certain level of fever and cough, most probably lungs X-ray and the SARS virus test should be involved for further accurate diagnosis. As a result, symptoms lungs X-ray and SARS virus test as well as DCG test become new attributes to the patients' existing dataset. Meanwhile, a number of new patients' records are appended continuously to the existing dataset as well. Under such a dynamic environment, learning must be intelligent and powerful enough to deal with any newly incoming data incrementally. Thus incremental learning is a basis for a robust and reliable diagnostic system.

On the other hand, there are few literatures discussing algorithms with incremental learning ability regarding the new attributes, even none is with respect to decision tree. And it is regarded as the most difficult and significant learning task [8]. Li et al. [9] proposed a new approach for constructing the concept lattice incrementally based on the increment of attributes, which resolves the updating problem of concept lattice due to the appended new attributes. Whereas an incremental neural network learning algorithm ILIA (Incremental Learning in terms of Input Attributes) with respect to the new incoming attributes is proposed in [10]. ILIA method retains the existing neural network while a new sub-network is constructed and trained incrementally when new input attributes are introduced, then the existing network and the new sub-network are merged to form a new final network for the changed problem.

Consequently, our novel learning algorithm  $i^+$ Learning (Intelligent, Incremental and Interactive Learning) is designed and proposed specifically for bridging the mentioned gaps. The philosophy behind it is simple but significant: a decision tree must grow automatically based on the existing tree model with respect to the new arriving instances as well as the new available attributes without retraining a new tree from scratch, where the new knowledge is learnt without forgetting the old one.

## 3. $i^+$ Learning algorithm

$i^+$ Learning theory is a new attempt that contributes the incremental learning community by means of intelligent, interactive, and dynamic learning architecture, which complements the traditional incremental learning algorithms in terms of performing knowledge revision in multiple dimensions. The algorithm grows an on-going decision tree with respect to either the new incoming instances or attributes in two phases: (1) Primary Off-line Construction of Decision Tree (POFC-DT): a fundamental decision tree construction phase in batch mode that is based on the existing database, where a C4.5-like decision tree model is produced; (2) Incremental On-line Revision of Decision Tree (IONR-DT): as incoming of the new instances or attributes, this phase is responsible for merging the new data into the existing tree model to learn incrementally the new knowledge by tree revision instead of retraining from scratch.

### 3.1. POFC-DT phase

This is an ordinary top-down decision tree construction phase that starts from the root node, using a splitting criterion to divide classes as "pure" as possible until a stopping criterion is met. The objective of this phase is to construct an optimal base tree, in order to have a robust foundation for further tree expansion. Binary tree structure is adopted in constructing such base tree. Binary tree has the same representational power as the non-binary tree, but it is simpler in structure and has no loss of the generated knowledge. This is because binary decision tree employs a strategy that a complex problem is divided into simpler sub-problems, in which it divides an attribute space into two sub-spaces repeatedly, with the terminal nodes associated with the classes [11].

To build a primitive binary tree, we start from a root node  $d$  derived from whichever attribute  $a_i$  in an attribute space  $A$  that minimizes the impurity measure. A binary partition can be denoted by a four-tuple representation  $(d, T, d^L, d^R)$ , where  $d$  is a decision node and  $T$  is a splitting

criterion on  $d$ , and  $d^L$  and  $d^R$  are the node labels for partitions of the left and right datasets respectively. Due to a binary tree is a collection of nested binary partitions, thus it can be represented in the following recursive form,

$$D = \{(d, SPL, d^L, d^R), D^L, D^R\} \quad (1)$$

where  $D^L$  and  $D^R$  denote the left and right sub-trees respectively, which are induced by the partition node  $d$  [12]. We employ Kolmogorov-Smirnoff (KS) distance [13], [14] as the measure of impurity at node  $d$ , which is denoted by  $I_{KS}(d)$  and is shown in equation (2) below:

$$I_{KS}(d) = \max_{v \in \text{value}(d)} (|F_L(v) - F_R(v)|) \quad (2)$$

where  $v$  denotes either the various values of a nominal attribute  $a$  with test criterion  $a = v$ , or a cut-point of a continuous-valued attribute  $a$  with test criterion  $a < v$ ;  $F_L(v)$  and  $F_R(v)$  are two class-conditional cumulative distribution functions that count the number of instances in the left and right sub-trees respectively, which is partitioned by a value  $v$  of an attribute  $a$  at a decision node  $d$ . KS is a well-known measure for the separability of two distribution functions, it is especially simple and computationally fast both in the training and classification stages. Hence, a best single test is picked across all attributes by enumerating the possible tests and selecting the one with the greatest KS distance. A decision tree grows by means of successive partitions until a terminal criterion is met.

### 3.2. IONR-DT phase

IONR-DT phase acts as a central character in our incremental decision tree algorithm. It embraces the faith that whenever a new instance and/or a new attribute is coming, this phase dynamically revises the fundamental tree constructed in POFC-DT phase without sacrificing the final classification accuracy, and eventually produces a decision tree as same as possible to those algorithms with all training examples available at the beginning. IONR-DT phase adopts the tree transposition mechanism that in ITI [16] as a basis to grow and revise the base tree. Besides, it preserves the essential statistical information to manage the decision tree. Such style of decision tree differs from the batch mode trees, since it remembers the information of instances regarding the respective possible values as well as the class label, in order to process the transposition without rescanning the entire dataset repeatedly.

KS measure is again applied in this phase for evaluating the goodness of a decision node. Once a (set of) new instance(s) is ready to be incorporated with an existing tree, IONR-DT phase carries out the following several steps: (1) updates the statistical information on each node that the

new instance traversed; (2) merges the new instance into an existing leaf or grows the tree one level under a leaf; (3) evaluates the qualification for the test on each node downwards starting from the root node; (4) for any attribute test that is no longer best to be on a node, the pull-up tree transposition process is called recursively to revise the existing decision tree; (5) finally, a new decision tree is revised and ready to perform the next classification.

### 3.3. $i^+$ Learning regarding attributes ( $i^+$ LRA)

Moreover, if an instance is available together with an unseen (new) attribute, except the above general steps, an additional procedure for incorporating a new attribute appropriately with an existing decision tree has to be called subsequently. In our algorithm, each new attribute has been treated at least medium important in default rather than noise in other algorithms, although its goodness measurement might be lower on its first occurrence. This is because in medical domain, a new symptom (attribute) has been involved into the original diagnostic rules usually implies that such symptom becomes a requisite condition in the subsequent diagnosis. Therefore, such attribute should logically be one of the decision nodes, even though the case is rare and the attribute might be irrelevant from the statistical point of view.

Further significant, in order to avoid the situation that an attribute has been appended mistakenly,  $i^+$ LRA offers the alternatives for a user to manually assign one of the four pre-defined importance grades to an attribute. This characteristic enables  $i^+$ LRA algorithm flexible enough to deal with the incremental data appropriately. Table 1 lists the meaning of each importance grade as well as the respective action being taken during the tree revision process.

Table 1. Four classes of the importance grades

Grade	Meaning	Action in Tree Revision
High	Very important, is a requisite condition in diagnosis.	Should be incorporated into the top several percentages of important decision nodes.
Medium	Important in the diagnosis of the most cases.	May not be a must, can be incorporated into the decision nodes above the average importance.
Low	Least important, can be ignored or promoted to higher level, depends on its supportive information.	Perhaps an irrelevant attribute that probably be appended closely to the leaf node and later on be pruned; or remain for supporting other attributes.
None	An attribute that is mistakenly added.	Treated as noise and ignored.

### 3.4. Importance measure ( $I_{ks}$ )

After selecting the importance grade from Table 1 for a new attribute, the crucial step is to determine a preliminary coefficient  $\mathcal{W}$  for its impurity measure ( $I_{KS}$ ). This coefficient is vital for a new attribute, and is used as a reference index for its importance measure. It is computed only once for each importance grade, in order to enable the new attribute to compete with its comparable attributes in being a decision node. This coefficient is decided as the ratio between the average KS measure of the attributes in the same rank that the new attribute belongs to, and the KS measure of the new attribute itself. Equation (3) illustrates the situation:

$$\mathcal{W} = \frac{\text{mean}\left(\sum_{i=1}^l I_{ks}(a_i)\right)}{I_{ks}(a_{new})} \quad (3)$$

where  $a_{new}$  is a new incoming attribute;  $a_i$  is an attribute of the same rank with  $a_{new}$ ; and  $l$  is the total number of such attributes, which is less than the total number of attributes in the attribute space  $A$ , i.e.  $l \leq |A|$ . Once such preliminary coefficient  $\mathcal{W}$  for  $a_{new}$  has been worked out, it could be applied to  $a_{new}$  by multiplying it to the KS measure of  $a_{new}$ , to enlarge the importance of  $a_{new}$  according to the given importance grade automatically. This strategy examines the initial importance of  $a_{new}$  regardless of its KS measure, which can prevent an actually important new attribute being treated as useless due to its newness and occupies little examples. Thereupon, a normal tree transposition process is carried on as usual to properly fit the new attribute  $a_{new}$  to a right position.

## 4. Experiments

### 4.1. Evaluation method

To realize the goodness of  $i^+$ Learning algorithm, the performance comparison of  $i^+$ Learning algorithms regarding the new instances only, and regarding the new instances as well as the new attributes are carried out against the non-incremental and incremental decision tree algorithms. C4.5 [15] and ITI algorithms are used as the representations of non-incremental and incremental learning families respectively. As before-mentioned, batch mode decision tree learning algorithms so far are able to guarantee the classification accuracy as well as the knowledge comprehensibility. Good incremental decision tree algorithms must be comparable to those non-incremental benchmark algorithms. C4.5 is one of the

most well-known state-of-the-art benchmark decision tree algorithms in batch mode. It is capable of yielding an optimal decision tree by using minimum information, which has better predictive power. In addition, C4.5 applied many enhancements over other algorithms, such as handling missing attribute values as well as continuous-valued attributes, and incorporates the tree pruning approach to avoid overfitting the data. The comparison between  $i^+$ Learning and C4.5 is able to manifest the effectiveness of  $i^+$ Learning method in dealing with multi-dimension incremental learning without sacrificing the learning performance.

For incremental learning family, only ITI algorithm is involved in the evaluation because ITI algorithm is a successor of ID5R, it performs incremental decision tree induction on symbolic or numeric variables, and handles noise and missing values. Moreover, ITI has been proven having the same effectualness as the tree induction algorithms in batch mode [16]. On the other hand, there is seldom algorithm analogous to  $i^+$ Learning except ITI, it stands on the same ground as  $i^+$ Learning although it is simply a unitary incremental methodology.

The evaluation of  $i^+$ Learning regarding attributes ( $i^+$ LRA) is more complicate and contains more procedures, which has been designed in the following several steps:

- 1) For each original training dataset  $D_{origin}$ , find out the root attribute  $A_{root}$  by performing  $i^+$ Learning in batch mode; where  $A_{root}$  will be treated as the new attribute being added later;
- 2) Divide  $D_{origin}$  into two portions:
  - A base training dataset  $D_{base}$  that excludes  $A_{root}$ , and contains only one-third instances of  $D_{origin}$ ;
  - An incremental training dataset  $D_{incr}$  that includes  $A_{root}$ , whereas  $A_{root}$  is appended at the end of the attributes list. It contains the remaining two-third instances of  $D_{origin}$ ;
- 3) Train  $D_{base}$  to construct a base classifier by performing  $i^+$ Learning algorithm;
- 4) Incorporate the new instances and  $A_{root}$  in  $D_{incr}$  with the base classifier produced on step 3). The algorithm incrementally and iteratively revises the base decision tree and eventually generates a final classifier;
- 5) Evaluate the final classifier generated on step 4) under the testing dataset, which is modified by moving  $A_{root}$  to the position just before class attribute.

The reason of selecting the root attribute  $A_{root}$  to be a new attribute being incorporated later, is because that  $A_{root}$  is the most informative attribute amongst all, whereas learning firstly without the most important attribute and

later has it back might be a best way to verify the ability of an incremental learning algorithm. On the other hand, the proportion of  $D_{base}$  to  $D_{incr}$  just follows the proportion of the benchmark testing and training datasets, which seems logical in the most cases.

#### 4.2. Evaluation result

Table 2 illustrates the result in classification accuracy that is evaluated on sixteen real-life datasets from UCI repository [17] over four learning algorithms. The learning algorithms are simply classifiers, having neither

pre-processing nor post-processing. This makes the corresponding comparisons much directly, native and pure. For clear comparison, the column entitled “Concl.” in the following table indicates the conclusion of the evaluation; while the symbols “●” or “--” are used to denote whether our algorithm (especially  $i^+$ LRA) is better (higher accuracy) or as best (almost same accuracy, the difference within 1%) as ITI algorithm; while the symbol “×” implies whether our algorithm is worse (smaller accuracy) than ITI. C4.5 algorithm used here is as an upper bound reference.

Table 2. Performance comparison in classification accuracy between various learning algorithms

Dataset	C4.5	ITI	$i^+$ Learning	$i^+$ LRA	Concl.
Cleve	75.2	65.347	71.287	81.188	●
Hepatitis	82.7	69.231	78.846	78.846	●
Hypothyroid	99.1	98.578	98.863	98.01	--
Heart	82.2	76.667	76.667	86.667	●
Sick-euthyroid	96.9	96.682	96.683	94.408	×
Auto	71	Error	68.116	57.971	●
Breast	94.4	94.421	95.279	95.708	●
Diabetes	69.1	66.797	73.047	73.438	●
Iris	92	94	94	94	--
Crx	82.5	76	82	84	●
Australian	86.1	77.391	81.304	88.261	●
Horse-colic	80.9	73.529	76.471	80.882	●
Mushroom	100	100	100	100	--
Parity5+5	50	50.586	51.758	51.563	--
Corral	90.6	100	100	90.625	×
Led7	67.4	63.367	63.567	64.767	●
<b>Average</b>	<b>82.506</b>	<b>75.162</b>	<b>81.743</b>	<b>82.521</b>	<b>●</b>

As revealed in the last row of the above table, our incremental algorithm  $i^+$ LRA has the identical average classification accuracy as the batch mode algorithm C4.5, which shows the promise that enhances the learning capacity without sacrificing the learning performance. On the other hand, the last column obviously demonstrated that  $i^+$ LRA algorithm (and  $i^+$ Learning as well) outperforms (either enhances or not degrades) ITI algorithm on fourteen datasets amongst all sixteen; whereas it degrades the classification accuracy on only two out of sixteen datasets. However, the difference of one downgraded dataset – *Sick-euthyroid* is only 2.274%, which is regarded as a sensible margin in general and is not considered as a change, where in many literatures are neglected. Besides, ITI is unable to build a decision tree for *Auto* (indicated as *Error*), which has numerical values in class attribute. These two experiments verify explicitly the robustness, the practicality,

the effectiveness and the superiority of  $i^+$ LRA over the state-of-the-art incremental learning algorithms. It is able to realize the learning in real time, incrementally and dynamically in multiple dimensions without sacrificing the learning performance.

#### 5. Conclusions

This paper has proposed a novel learning algorithm  $i^+$ Learning as well as  $i^+$ LRA, which apparently achieves the highest classification accuracy over ITI algorithm. The solid evidence manifests that  $i^+$ Learning as well as  $i^+$ LRA do superior to other incremental learning algorithms not only on the classification accuracy, but also be able to handle the incremental learning regarding the new incoming attribute other than the new instance only without sacrificing the learning performance. Such results bring out

the following significant view:  $i^+$ Learning can successfully mimic the learning style in real world, which is real time and dynamic in multiple dimensions that includes both new input attributes and instances. In addition, the incremental learning strategy is able to accelerate the training time and meanwhile new knowledge can be accumulated or revised without forgetting the old one.

However, there is no perfect algorithm, which is also true to  $i^+$ Learning. The major limitation of our method is the adoption of binary tree rather than multi-branch tree. Such structure increases the tree size, whereas an attribute can be selected as a decision node for more than once in a tree. For that reason, binary trees tend to be less efficient in terms of tree storage requirements and test time requirements, although they are easy to build and interpret.

In the future work of our research, it would be valuable that  $i^+$ Learning model can be extendable for classifying multi-label class problem, in which an instance belongs to multiple classes simultaneously [18]. Moreover, the incremental learning method with respect to new output classes in addition to instances and attributes is another influential consideration in future  $i^+$ Learning model.

## Acknowledgements

The authors are grateful to the Faculty of Science and Technology of the University of Macau for supporting our research in various aspects.

## References

- [1] D. Michie, "Machine Learning and Knowledge Acquisition", International Handbook of Information Technology and Automated Office Systems, Elsevier Science Publishers, North-Holland, 1985.
- [2] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer-Verlag, New York, 2001.
- [3] I. Kononenko, "Inductive and Bayesian Learning in Medical Diagnosis", Applied Artificial Intelligence, Vol. 7, pp. 317-337, 1993.
- [4] P.E. Utgoff, "ID5: An Incremental ID3", Proceedings of the 5th International Conference on Machine Learning, Ann Arbor, MI, pp. 107-120, 1988.
- [5] P.E. Utgoff, "Incremental Induction of Decision Trees", Machine Learning, Vol. 4, No. 2, pp. 161-186, 1989.
- [6] P.E. Utgoff, "An Improved Algorithm for Incremental Induction of Decision Trees", Proceedings of the 11th International Conference on Machine Learning, Morgan Kaufmann, New Brunswick, NJ., pp. 318-325, 1994.
- [7] D. Kalles, and T. Morris, "Efficient Incremental Induction of Decision Trees", Machine Learning, Vol. 24, No. 3, pp. 231-242, 1996.
- [8] Z.H. Zhou, and Z.Q. Chen, "Hybrid Decision Tree", Knowledge-Based Systems, Vol. 15, pp. 515-528, 2002.
- [9] Y. Li, Z.T. Liu, L. Chen, X.J. Shen, and X.H. Xu, "Attribute-based Incremental Formation Algorithm of Concept Lattice", Mini-micro Systems, Vol. 25, No. 10, pp.1768-1771, 2004.
- [10] S.U. Guan, and S.C. Li, "Incremental Learning with Respect to New Incoming Input Attributes", Neural Processing Letters, Vol. 14, pp. 241-260, 2001.
- [11] H.R. Bittencourt, and R.T. Clarke, "Use of Classification and Regression Trees (CART) to Classify Remotely-Sensed Digital Images", Proceedings of International Geoscience and Remote Sensing Symposium, IGARSS '03, Vol. 6, pp. 3751-3753, 2003.
- [12] S.R. Safavian, and D. Landgrebe, "A Survey of Decision Tree Classifier Methodology", IEEE Transactions on Systems, Man and Cybernetics, Vol. 21, No. 3, pp. 660-674, 1991.
- [13] J.H. Friedman, "A Recursive Partitioning Decision Rule for Nonparametric Classification", IEEE Transactions on Computers, C-26, pp. 404-408, 1977.
- [14] P.E. Utgoff, and J.A. Clouse, "A Kolmogorov-Smirnov Metric for Decision Tree Induction", Technical Report 96-3, Amherst, MA: University of Massachusetts, Department of Computer Science, 1996.
- [15] J.R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993.
- [16] P.E. Utgoff, N.C. Berkman, and J.A. Clouse, "Decision Tree Induction Based on Efficient Tree Restructuring", Machine Learning, Kluwer Academic Publishers, Vol. 29, pp. 5-44, 1997.
- [17] C.L. Blake, and C.J. Merz, "UCI Repository of Machine Learning Databases", Department of Information and Computer Science, University of California, 1998. [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]
- [18] Y.L. Chen, C.L. Hsu, and S.C. Chou, "Constructing a Multi-Valued and Multi-Labeled Decision Tree", Expert Systems with Applications, Vol. 25, pp. 199-209, 2003.