

An Efficient Visual Classification Based Approach to Decision Tree Construction

Tahmina Rahman and Koichi Harada

Department of Information Engineering, Hiroshima University, Japan.

d101711@hiroshima-u.ac.jp, hrd@hiroshima-u.ac.jp

Abstract— Decision Tree (DT) analysis has emerged over the decades as an effective tool in classification or prediction. Since the publication of the first comprehensive and authoritative book on decision tree analysis by Howard Raiffa in 1968, its applications to a variety of problems from numerous disciplines have grown enormously. However, most of the methods for DT construction have some pitfalls including binary split points of numeric attributes instead of arbitrary splitting, involvement of users with prior domain knowledge to construct DT and finally the absence of training data visualization which this paper aims to remove. Besides we have proposed how the constructed DT can be applied for the adaptive interview process.

Keywords— Decision Tree, Adaptive interview, Response Analysis, Data visualization etc.

I. INTRODUCTION

Decision tree analysis has emerged over the decades as an effective tool in classification or prediction. A decision tree is particularly helpful in situations of complex multistage decision problems. For example, when we need to plan and organize a sequence of decisions and take into account how the choices made at earlier stages and the outcomes of possible external events determine the types of decisions and events at later stages of that sequence. A decision making tree is essentially a diagram that represents, in a specially organized way, the decisions, the main external or other events that introduce uncertainty, as well as possible outcomes of all those decisions and events. The appeal of decision trees is due to the fact that, in contrast to other methods, decision tree represents rules. These rules can readily be expressed so that humans can understand them or even directly used in applications. In some of these applications, the accuracy of classification or prediction is the only thing that matters. In such cases it is not a question how or why the model works. In some other situations, the ability to explain the reason for a decision is decisive.

There are many algorithms for building decision trees providing desirable quality of interpretability. A frequently used method over the years is C4.5 [1]. This and other most commonly used decision tree construction algorithms perform a binary split of the form $a \leq v$ for a numeric attribute a and a real number v . The SPRINT decision tree classifier [2] processes numeric attributes in the following way. There are $n - 1$ possible splits for n distinct values of a . The *gini* index is calculated at each of these $n - 1$ points and the attribute value providing the lowest *gini* index is taken as the split point.

CLOUDS [3] draws a sample from the set of all attribute values and evaluates the *gini* index only for this sample which in turn improves the efficiency. A commercial system namely SPSS CHAID [4] for interactive decision tree construction does not visualize the training data but only the decision tree. Besides, the interaction happens only before the tree construction yielding user defined values for global parameters such as maximum tree depth or minimum support for a node of the decision tree. In the recent years, visual representation of data as a basis for the human computer interface has evolved rapidly. The paper [5] provides a comprehensive overview over existing visualization techniques for huge multidimensional data. Recently, several techniques of visual data mining have been introduced. The research paper [6] presents the technique of Independence Diagrams for visualizing dependencies between two attributes. The brightness of a cell in the two-dimensional grid is set proportional to the density of corresponding data objects. This is one of the few techniques which do not visualize the discovered knowledge but the underlying data. The authors of the paper [7] provide a decision table classifier and a method of visualization the resulting decision tables. It is argued that the visualization is appropriate for business users who are unfamiliar with machine learning concepts. Almost all of these methods have the pitfalls which primarily include the binary split points of numeric attributes instead of arbitrary splitting, must have engaged users with prior domain knowledge and finally the absence of training data visualization and the option of backtracking.

This paper presents a technique that enables arbitrary split points for numeric attributes, the novice user's capability of constructing the decision tree with the option of backtracking, the visualization of training data during the construction of the decision tree with higher accuracy and lower tree size compared to different existing methods. The way how this DT can effectively be applied to the adaptive interviewing and response analysis has also been presented.

The remaining of our paper is composed in the following ways. The next section represents an example how we construct a decision tree. Section 3 presents the ways how the split points can be interactively selected which actually ultimately facilitates the novice users to build DT. The next section 4 demonstrates the way to visualize the training data. Section 5 describes the procedure to apply the constructed DT for adaptive interviewing processing along with the response analysis.

II. CONSTRUCTION OF DECISION TREE

Let us consider the learning task represented by the training example of Table 1. Here the target attribute *PlayTennis*, which can have values *yes* or *no* for different days, is to be predicted based on other attributes of the day in the questions.

Table 1: TRAINING EXAMPLES FOR *PLAYTENNIS*

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Now let us see how to construct a decision tree from the training data in Table 1. The central choice in the algorithm is selecting which attribute to test at each node in the tree. We would like to select the attribute that is the most useful for classifying examples. What is a good quantitative measure of the worth of an attribute? It defines a statistical property *information gain* that measures how well a given attribute separates the training examples according to their target classification. It uses this *information gain* measure to select among the candidate attributes at each step while growing the tree.

To define the *information gain* precisely, the concept of entropy is employed. Given a collection S , containing positive and negative examples of some target concept, the entropy of S relative to this boolean classification is-

$$Entropy(S) = -p_{+ve} \log_2 p_{+ve} - p_{-ve} \log_2 p_{-ve}$$

where p_{+ve} is the proportion of positive examples in S and p_{-ve} is the proportion of negative examples in S .

To illustrate, in the examples we consider (Table 1) S is a collection of 14 examples of some boolean concept, 9 of which are positive examples (attribute *PlayTennis*) and 5 are negative (attribute *PlayTennis*). We denote it by the notation $[9+, 5-]$. Then the entropy of S relative to this boolean classification is-

$$Entropy([9+, 5-]) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.94$$

More generally, if the target attribute can take on c different values, then the entropy of S relative to this c -wise classification is defined as

$$Entropy(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

The measure *information gain* is simply the expected reduction in entropy caused by partitioning the examples

according to this attribute. More precisely, the *information gain*, $Gain(S, A)$ of an attribute A , relative to a collection of examples S , is defined as-

$$Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where $Value(A)$ is the set of all possible values for attribute A , and S_v is the subset of S for which attribute A has value v (i.e. $S_v = \{s \in S \mid A(s) = v\}$). Note that the first term of the above equation is just the entropy of the original collection S , and the second term is the expected value of the entropy after S is partitioned using attribute A .

Suppose we want to calculate the *information gain* of the attribute *Wind* of the example. This attribute can have the values *Weak* and *Strong*. As before, here S is the collection of 14 examples $[9+, 5-]$. Of these examples, 6 of the positive and 2 of the negative examples have *Wind=Weak*, and remainder have *Wind=Strong*. The *information gain* due to sorting the original 14 examples by the attribute *Wind* ($S=[9+, 5-]$, $S_{Weak}=[6+, 2-]$, $S_{Strong}=[3+, 3-]$) can be calculated as-

$$\begin{aligned} Gain(S, Wind) &= Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\ &= Entropy(S) - \left(\frac{8}{14}\right) Entropy(S_{Weak}) - \left(\frac{6}{14}\right) Entropy(S_{Strong}) \\ &= 0.94 - \left(\frac{8}{14}\right) 0.811 - \left(\frac{6}{14}\right) 1.00 = 0.048 \end{aligned}$$

If we calculate the *information gain* of all the four attributes (*Outlook*, *Humidity*, *Wind* and *Temperature*) we find that $Gain(S, Outlook)=0.246$, $Gain(S, Humidity)=0.151$, $Gain(S, Wind)=0.048$ and $Gain(S, Temperature)=0.029$.

According to the *information gain* measure, the *Outlook* attribute provides the best prediction of the target attribute, *PlayTennis*, over the training examples. Therefore, we have to select *Outlook* as the decision attribute for the *root* node, and the branches are created below the root for each of its possible values (*Sunny*, *Overcast*, *Rain*). The resulting partial decision tree is shown in the Figure 1.

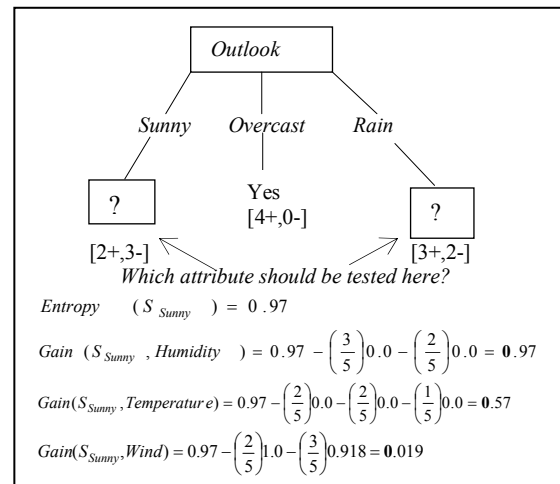


Figure 1: A partially learned decision tree

As we see, if the *Outlook* is *Sunny* then we have tested the attribute *Humidity* as it has the higher *information gain* (0.98) relative to *Temperature* (0.57) and *Wind* (0.019).

The process of selecting a new attribute and partitioning the training examples is now repeated for each nonterminal descendant node, this time using only the training examples associated with that node. This process continues for each new leaf node until either of the following two conditions is satisfied:

1. Every attribute has already been included along this path through the tree.
2. The training examples associated with this leaf node all have the same target attribute value (i.e. their entropy is zero).

The complete decision tree for the considered example is shown in the Figure 2 [3].

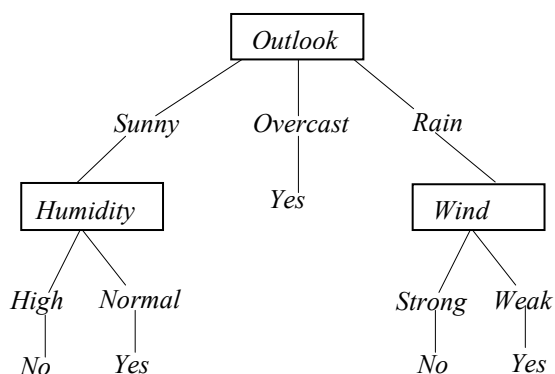


Figure2: A decision tree for the concept of *PlayTennis*

In general, decision tree represents a disjunction of conjunctions of constraints on the attribute values of instances. Each path from the tree root to a leaf corresponds to a conjunction of attribute tests and the tree itself to a disjunction of these conjunctions. For instance, the decision tree shown in Figure 2 corresponds to the expression:

$$(Outlook = Sunny \wedge Humidity = Normal) \vee (Outlook = Overcast) \vee (Outlook = Rain \wedge Wind = Weak)$$

III. INTERACTIVE SPLIT POINTS SELECTION

The proposal selects split points interactively which can be done in two steps: (a) deciding split lines and (b) selecting a split point on each of the decided split lines. At first, the user selects a split line which is one of the lines (orthogonal to the segment halving line) upon which the pixels are arranged, by clicking on any pixel in the chosen segment. Then this split line is replaced with an animated line on which alternatively black and white strips move along. As the black and white colors are not used to map the classes, the brushed split line will be well perceptible. The pixels of the selected split line are redrawn in a separate area in a magnified fashion enabling the user to set the exact split point.

Splitting Criteria

As stated earlier that most of the decision tree construction methods uses binary splits. But here we overcome this limitation of binary splits in attributes with a continuous

domain. This extra flexibility raises the question about an appropriate splitting strategy. The first of the four options stated below is chosen which is applicable in the current visualization. The term ‘partition’ is used for a coherent region of attribute values in the splitting attribute that the user wants to separate by split points.

i) First choose the segment with the largest pure partitions. A partition is called ‘pure’ if the user decides to label this partition with the most frequent class. This decision leads to leaf nodes in the decision tree, thus reducing the size of data which is not classified.

ii) The segment with the largest cluster clearly dominant in one colour should be chosen, if no pure partition is perceptible. In contrast to a pure partition, such a cluster will not be labelled by the most frequent class.

iii) If a choice upon A or B fails, the segment should be chosen that contains the most pixels that can be divided into partitions where each has one clearly dominant colour.

iv) If none of the above options applies, choose the segment where different distributions can be best separated through partitioning.

IV. VISUALIZATION OF TRAINING DATA

The training data is visualized in order to facilitate the interactive decision tree construction. It is possible to visualize multi-dimensional data with a class label such that their degree of impurity with respect to class membership can be easily perceived by a user. We have used the pixel-oriented method which maps the classes to colours in an appropriate way. The primary idea comes from [5], [8]. It maps each attribute value v_i of each data object to one coloured pixel and represents the values belonging to different attributes in separate sub-windows. The method maps d -dimensional objects to a circle which is partitioned into d segments representing one attribute each. Within each segment, the arrangement starts in the middle of the circle and continues to the outer border of the corresponding segment in a line-by-line mode. These lines upon which the pixels are arranged are orthogonal to the segment halving lines. In order to map each attribute value of D to a unique pixel, the idea of the Circle Segments technique has been followed. However, we need not necessarily use the overall distance from a query to determine the pixel position of an attribute value. Rather, in the proposed method, we sort each attribute separately and use the induced order for the arrangement in the corresponding circle segment. The colour of a pixel is determined by the class label of the object to which the attribute value belongs. In the proposed approach of visualization of training data, attributes having small number of distinct values are considered. Thus, different objects sharing same attribute value is not uniquely defined. Depending on this chosen order, we can create homogeneous

(with respect to the class label) areas within the same attribute value.

V. ADAPTIVE INTERVIEWING AND RESPONSE ANALYSIS

Adaptive Interview

The DT constructed by the proposed method with the arbitrary splitting points and data visualization is targeted to be applied of adaptive interviewing for the E-commerce purpose. The proposed structure of the decision tree for the adaptive interview can be shown in Figure 3.

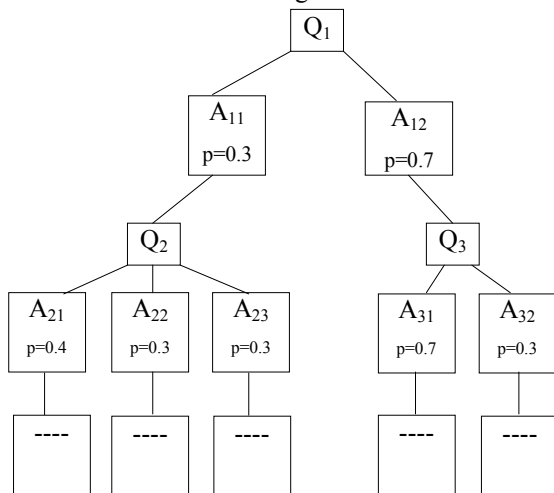


Figure 3: Structure of DT for adaptive interview

Here any interviewee can answer A_{11} or A_{12} to the question Q_1 . According to the notion of the decision tree, the question Q_2 follows the answer A_{11} and the question Q_3 follows the answer A_{12} and so on. For the interviewing process, the foremost task is to determine a set of questions related to finding out the target answers of the system the approach is going to be applied to. The possible answer set to each of these questions is also determined. To do this, the system is to be analysed thoroughly so that no questions and the possible answers to the questions remain excluded.

In the second step, we collect the set of training data that facilitates having an assumption for the user about the behaviours of the interviewees. The more the training data, the more the accuracy of the tree optimization is.

The third step is to set the goal criterion. It expresses how well the currently considered path sequence in the tree matches the preferences of the interviewees and the interviewers. In the tree optimization, a set of dependent probabilities are used. Some cases arise where there is considerable difference between the dependent probability and the total probability. In this case, the variance for the set of dependent probabilities for each question is calculated. The total probability is chosen as a good approximation of the dependent probability when the variance is low. When the variance is high then the total probability is a defective approximation of the real dependent probability.

Response Analysis

Response analysis of each of the question is a very important step in building the decision support systems. However, there are many factors in the response analysis such as the probability of each answers, the importance of each answers of the interviewee etc. The improvement of the decision analysis partially depends on the improvement of other models in the decision support systems. Some results of other models in the decision support systems can serve as the inputs for the response analysis. Thus, the importance of the response analysis of the interviewing process for a specific problem is, no doubt, very high. Barrett and Maxwell [9] advanced a decision tree to guide the response analysis, which was adapted by various agencies and authors [10] [11]. While the paper [9] intended this Decision Tree to reflect the decision making during emergencies, it is equally applicable to non-emergencies. The main difference between emergency and non-emergency programming is the time frames within which analysts must operate and the relative predictabilities of the non-emergency and emergency impacts on markets and households. In emergencies, we have to prioritize the data collection and analysis based on data availability and their understanding of local market functioning. In the proposed research, the method [9] with some modifications is applied in order to fit into the problem domain of E-commerce, namely the adaptive interview process.

VI. DISCUSSION

1. It might be the case that in the initial tree, one question is asked in several locations in the decision tree. In this case, a set of dependent probabilities is found for this question which in some cases, might differ significantly from the total probability. For example, for a cluster of buyers, which answer question X with the answer X_1 , the probability of then answering Y with Y_2 is much higher than for those who answered X with Y_2 . It may also be possible to assess for each question the variance for the set of dependent probabilities. When this variance is small, the total probability can be used as a good approximation of the dependent probability. But, in the case where the variance is high, total probability might be an unreliable approximation of the real dependent probability. In such case, we can have more data to properly assess these dependent probabilities. On the basis of the initial iterative optimization technique such data is missing because some tree traversals (i.e. sequences of questions) will never be offered to users owing to the fact that the assumed probabilities of question answers indicate the optimality of other question sequences. In such circumstance, the highly ranked questions are asked. The result of such an optimization is a probabilistic decision tree which specifies for each answer given not only one, but a number of possible next questions that should be asked with a given probability. It thus allows the collection of actual answer probabilities for a wider range of question sequences and, hence, provides a better basis for the continuous optimization of

the decision tree.

2. It is the fact that in some e-commerce systems the initial set of products to be considered in the interview process is not the same for all buyers because there is the parallel use of other techniques like pre-filtering (e.g. through shopping agents) or personalization. In the worst case, an individual buyer starts with a different set of products, and in some systems this set of products may also change independently of the information collected in the interview. It will not be possible to pre-compile the decision trees in a "batch" procedure in such case. So, the reasoning about the next best question is to be done dynamically based on the available data. Such online reasoning might place difficult real-time constraints on the algorithms employed. As a result, exploring mechanisms for partial compilation and other optimization methodologies targeted at the algorithm's performance should be constructed with much cares and considerations.
3. A very common problem associate with e-commerce system is that some users visiting shops may exit before they actually purchase something. Therefore one of the important characteristics of a good e-commerce system is that it reduces the number of user exits and finally turns browser into buyer. We can achieve this by reducing the number of questions necessary to determine a meaningful decision, which can be presented to the user. A meaningful stop criterion set in the method of the decision rule in the tree adjustment step is one element supporting this strategy, as it reduces the number of questions by preventing useless questions in the sense that these questions do not increase the expected target value. The advanced optimization of the number of questions might be achieved when the probabilities of consecutive answers are reflected in the selection of the next questions.

VII. CONCLUSIONS

The proposed method has enabled the multiple splitting points and the visualization of the training data to facilitate the novice user to construct the decision tree interactively. The

constructed DT is applied for the adaptive interview process which is very effective at the time of purchasing the product in the E-commerce system. The method is useful in the sense that it provides a simple and effective criterion for splitting the tree. This can be extended to handle more complex optimization requirements. The change/update of the interview process as well as the change of the behaviours of the interviewees can be effectively managed through the regeneration of the interview decision tree. At present the system can handle only one-to-one interviewer-interviewee relationship. This can be extended to one-to-many or even many-to-many relationship as the future work.

REFERENCES

- [1] The C4.5 method for Decision Tree Construction.
- [2] Agrawal R., Mehta M. and Shafer J.C., "SPRINT: A Scalable Parallel Classifier for Data Mining", in Proc. VLDB '96, 22nd Intl. Conf. on Very Large Databases, Bombay, India, 1996, pp. 544-555.
- [3] Alsabti K., Ranka S. and Singh V., "CLOUDS: A Decision Tree Classifier for Large Datasets", in Proc. KDD '98, 4th Intl. Conf. on Knowledge Discovery and Data Mining, New York City, 1998, pp. 2-8.
- [4] The website: <http://www.spss.com/>.
- [5] Keim D. A., "Visual Database Exploration Techniques", Proc. Tutorial Int. Conf. on Knowledge Discovery & Data Mining, Newport Beach, CA, 1997.
- [6] Berchtold S., Jagadish H.V. and Ross K.A., "Independence Diagrams: A Technique for Visual Data Mining", in Proc. KDD '98, 4th Intl. Conf. on Knowledge Discovery and Data Mining, New York City, 1998, pp. 139-143.
- [7] Kohavi R. and Sommerfield D., "Targeting Business Users with Decision Table Classifiers", in Proc. KDD '98, 4th Intl. Conf. on Knowledge Discovery and Data Mining, New York City, 1998, pp. 249-253.
- [8] Ankerst M., Keim D. A. and Kriegel H.-P., "Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets", Proc. Visualization '96, Hot Topic Session, San Francisco, CA, 1996.
- [9] Christopher B. Barrett and Daniel G. Maxwell, "Food Aid After Fifty Years: Recasting its Role", London: Routledge, 2005.
- [10] Pantaleo Creti and Susanne Jaspers, eds., "Cash Transfer Programming in Emergencies", Oxfam GB, Oxford, 2006.
- [11] Ugo Gentilini, "Cash and Food Transfers: A Primer", World Food Programme Occasional Paper no. 18, 2007.