Revista Ingeniería Industrial UPB / Vol. 04 / No. 04 / pp. 17-24
enero-diciembre / 2016 / ISSN: 2346-2299 / 2357-6839 (En línea) / Medellín- Colombia

Comparación de programas computacionales para árboles de decisión

Comparison of decision tree computer programmes

Moritz Albrecht
moritzalbrecht@outlook.de

Javier Darío Fernández Ledesma
javier.fernandez@upb.edu.co

Escuela de Ciencias Estratégicas y Escuela de Ingenierías, Universidad Pontificia Bolivariana Medellín, Colombia.

Los árboles de decisión son herramientas poderosas para analizar posibles salidas en las decisiones óptimas.
Los árboles de decisiones pueden ser muy complejos y el software usado que se propone permite el análisis de una manera simple; este es el propósito del artículo: analizar las diferencias entre los programas existentes en el mercado.

PALABRAS CLAVE
Decision, Programas

RESUMEN

ABSTRACT

Decision Trees are a helpful tool to analyse possible outcomes from decision and to decide upon the optimal one. As decision trees can be very complex and purpose used software allows easy further analysis the article analyses the differences between on the market programmes.

KEYWORDS
Decision, Programmes.

I. Introduction

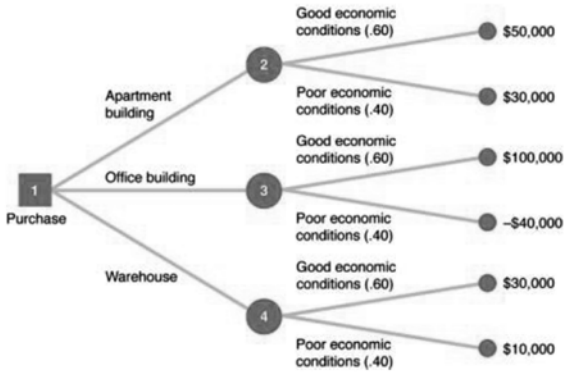
Decision Trees depict transparently a complex, if necessary multilevel decision making process with all possible decision options. Interlinked decisions can be visualized via ramifications and decisions can stand in a temporal or logical sequence. Normally decision trees can be of a logical or mathematical nature. Although decision trees can be of a logical or a mathematical nature, obviously programmes will make use of mathematical algorithms. Overall decision trees can be used in complex situations of decisions and for analysis of problems or issues through a systematic and structural description of all decision options. Prerequisites are estimates of probabilities of occurrence of the options and that there is a manageable amount of options [1].

There are 4 steps for creating a decision tree:

- Creation of the tree structure: A decision tree begins with the first decision shown through a small square from which lines connect to at least two possible options or solutions. If a solution is unknown this can be shown by a small circle demonstrating an uncertainty. Further decision will as well be shown through small squares as well as uncertainties through small circles. If a branch terminates this can be shown through a line terminated by a triangle or no symbol at all.
- Evaluation of branches: In this step each branch has to be evaluated by its potential if applicable financial result. Otherwise you can assess which investment you would realize for archiving this result.
- Evaluation of probabilities: If uncertain solution arise you most probably have different possible scenarios as solution, which can be evaluated by probability of occurrence, where the sum of these at each not has to equal 100%. If no probabilities exist, they have to be guessed.
- Calculation of tree values: Beginning at the right the value for each knot is to be calculated and evaluated by comparison of all options.[1]

The following shows an extremely simplified drawn decision tree. The main decision, shown on the outer left is the purchase of either an apartment building, an office building or a warehouse. Depending on the unfortunate event of the economic conditions, in this case poor and good with a probability of 60 and 40 per cent, each decision will result in two possible overall outcomes shown on the outer right. The investments of each decision and each earnings are not shown, but the optimum decision is the one with the highest roll-back value. This one is calculated by the cross product of the probabilities and each outcome of the unfortunate event for each decision. For example the rollback value for investing in the warehouse is the result the sum of the multiplications of the outcomes of 30000 and 10000 with its probabilities of 0.6 and 0.4: 30000$*0.6+10000$*0,4= 22000 $. This is the worst decision as the expected value for the apartment building is 42000 $ and for the office building 44000$. With that the optimal decision is the purchase of the office building and the best possible outcome would be 100000$.

Figure 1. Simplified decision tree example [2]



This is just an example of a simplified decision tree problem. Complex correctly created decision trees should be simulated by computer programmes.

REMARK: Although decision trees represent a very promising and popular approach for mining data, it is important to note that this method also has its limitations. The limitations can be divided into two categories: (a) algorithmic problems that complicate the algorithm's goal of finding a small tree and (b) problems inherent to the tree representation. Top-down decision-tree induction algorithms implement a greedy approach that attempts to find a small tree. All the common selection measures are based on one level of look ahead. Two related problems inherent to the representation structure are replication and fragmentation. The replication problem forces duplication of sub-trees in disjunctive concepts, such as (A∩B)U(C∩D) (one sub-tree, either(A∩B)or(C∩D) must be duplicated in the smallest possible decision tree); the fragmentation problem causes partitioning of the data into smaller fragments. Replication always implies fragmentation, but fragmentation may happen without any replication if many features need to be tested. This puts decision trees at a disadvantage for tasks with many relevant features. More important, when the datasets contain large number of features, the induced classification tree may be too large, making it hard to read and difficult to understand and use. On the other hand, in many cases the induced decision trees contain a small subset of the features provided in the dataset [3].

The basic concept behind the decision tree classification algorithm is the partitioning of records into "purer" subsets of records based on the attribute values. A pure subset is one in which all the records have the same class label. The end result of the decision tree algorithm is the output of classification rules that are simple to understand and interpret. This interpretability property is strength of the decision tree algorithms. In general, these algorithms find the attribute that best splits a set of records into a collection of subsets with the greatest overall purity measure. The purity of a subset can be quantified by entropy, which measures the amount of information loss. Entropy is a real number between zero and one, where an entropy value of zero indicates that the data set is perfectly classified while a value of one indicates that no information has been gained. The algorithm recursively operates on each newly generated subset to find the next attribute with which to split the data set. The algorithm stops when all subsets are pure or some other stopping criterion has been met. Examples of stopping criteria include the exhaustion of attributes with which to split the impure nodes, predefined node size, and minimum purity level.

An up-to-date algorithmic framework for top-down induction of decision trees is presented in the following figure 2. It contains three procedure; one for growing the tree (TreeGrowing), one for pruning the tree (treePruning) and one to combine those two procedure (inducer):

Figure 2. Generic algorithmic framework for top-down induction of decision trees. Inputs are the training set X, the predictive attribute set A and the target attribute [4].

```
1: procedure inducer(X, A, y)
2:     T = treeGrowing(X, A, y)
3:     return treePruning(X, T)
4: end procedure
5: procedure treeGrowing(X, A, y)
6:     Create a tree T
7:     if one of the stopping criteria is fulfilled then
8:         Mark the root node in T as a leaf with the most common value of y in X
9:     else
10:        Find an attribute test condition f(A) such that splitting X according to f(A)'s outcomes (v_1, ..., v_l) yields
           the best splitting measure value
11:        if best splitting measure value > threshold then
12:            Label the root node in T as f(A)
13:            for each outcome v_i of f(A) do
14:                X_{f(A)=v_i} = {x ∈ X | f(A) = v_i}
15:                Subtree_i = treeGrowing(X_{f(A)=v_i}, A, y)
16:                Connect the root node of T to Subtree_i and label the corresponding edge as v_i
17:            end for
18:        else
19:            Mark the root node of T as a leaf and label it as the most common value of y in X
20:        end if
21:    end if return T
22: end procedure
23: procedure treePruning(X, T)
24:    repeat
25:        Select a node t in T such that pruning it maximally improves some evaluation criterion
26:        if T ≠ ∅ then
27:            T = pruned(T, t)
28:        end if
29:    until T = ∅ return T
30: end procedure
```

II. Excel add-in or standalone software?

Before deciding on which programme to use, it should be distinguished between the pros and cons of excel add-ins and standalone softwares. An Excel Add-In is a prewritten VBA Extension

for Microsofts table-calculations-software Excel. **V**isual **B**asic for **A**pplications is an event drive programming language from Microsoft for all its Office applications. The advantage of using VBA is the possibility of debugging and changing the macro input and with the possibly optimizing the prewritten programme. But all liable to costs programmes treated in this article are protected by a ("crackable") password. Anyways they are quite extent and perfect already not needing much of a makeover. The most positive thing about using Add-ins is the familiarity with Excel itself and the possibilities it gives. There are not many cons of using MS Excel Add-ins: The first one is that you need to know how to activate and use the add-ins properly and the second one is the restricted VBA language.

Using a standalone software means installing another programme which is not editable. But this means on the other hand the gaining of getting rid of Excel restrictions and extend the possibilities of decision finding. Whereas this makes it less comprehensive for layman.

III. Excel add-ins

A. PrecisionTree

PrecisionTree is a programme provided by Palisade. It provides a formal structure in which decisions and chance events are linked in sequence from left to right. Decisions, chance events, and end results are represented by nodes and connected by branches. The result is a tree structure with the "root" on the left and various payoffs on the right. Probabilities of events occurring and payoffs for events and decisions are added to each node in the tree. With PrecisionTree, it is possible to see the payoff and probability of each possible path through a tree. PrecisionTree functions may be added to any cell in a spreadsheet and can include arguments that are cell references and expressions - allowing great flexibility in defining decision models. It is also to collapse and restore branches to the right of any given node for simplicity and easier navigation through the tree, and insert nodes at any point in a tree. As well it is possible to even append symmetric subtrees to particular nodes, greatly speeding up the building of large models. PrecisionTree determines the best decision to make at each decision node and marks the branch for that decision TRUE. Once your decision tree is complete, PrecisionTree's decision analysis creates a full statistics report on the best decision to make and its comparison with alternative decisions. PrecisionTree can create a Risk Profile graph that compares the payoffs and risk of different decision options. It displays probability and cumulative charts showing the probabilities of different outcomes and of an outcome less than or equal to a certain value. PrecisionTree can also perform a sensitivity analysis by modifying the values of the variables you specify and recording the changes in the expected value of the tree. It is possible to change one or two variables at a time. Results include sensitivity, tornado, spider, and strategy-region graphs.

As well PrecisionTree includes advanced features like Bayesian Revios in order to "flip" one or more chance nodes in a model in order to show probabilities calculated using Bayes' Rule. This is valuable when the probabilities of a model are not available in a directly useful form. For example, you may need to know the probability of an outcome occurring given the results of a particular test. The test's accuracy may be known, but the only way to determine the probability you seek is to "reverse" a traditional tree using Bayes Rule. Another feature are Logic nodes, where the optimum branch is selected according to conditions the user defines. A logic node behaves like a decision node, but it selects the branch whose branch logic formula evaluates to TRUE as the logical (optimum) decision. Another feature are reference nodes which enable to reference to a sub-tree. The sub-tree can be on any sheet in the workbook. Use reference nodes to simplify a tree, to reference the same sub-tree many times in a tree, or to build a tree that's too large to fit on one spreadsheet. It is also possible to use Linked Trees, which allow the branch values for a decision

tree to be linked to cells in an Excel model external to the tree. Each node can be linked to an Excel cell reference or range name. End node payoffs can be calculated by a detailed spreadsheet model. This powerful feature combines the strength of a decision tree for describing decision situations with the strength of a traditional spreadsheet model for calculating results. Another feature is the possibility to calculate decision tree path payoff values using a custom VBA formula and with that having the possibility to drastically simplify models. As well the user can use Custom Utility Functions by standard PrecisionTree calculations or is even possible to modify them using macros. Another advanced features is the Developer Kit, a built-in programming language that allows you to automate PrecisionTree using Excel VBA. The last additional feature are Influence Diagrams **which are u**sing nodes and arcs, influence diagrams are used to summarize the general structure of a decision and they can also represent asymmetric trees.

PrecisionTree is available by itself or as part of the DecisionTools Suite which includes @ RISK, adding risk analysis to Excel using Monte Carlo simulation, TopRank for what-if analysis, NeuralTools and StatTools for data analysis, and other. The pro of the DecisionTools Suite is that components are compatible with each other can be combined for greater insight and analysis. For example: PrecisionTree represents chance events and payoffs with discrete values or probabilities. When combined with @RISK, @RISK enhances the analysis by representing continuous ranges of outcomes for chance events and payoffs. By running Monte Carlo simulation on your decision tree, it is possible to see more possible outcomes of a decision. (See Fig. 3)

B. SolutionTree

SolutionTree is an add-in provided by StricklenSolutions. It includes like PrecisionTree and Easy-to.use Ribbon and provides all basic features for creaing, copying and collapsing. As well you can change basic settings like calculation methor or format belong others. As well you can run automatic reports like the optimal policy tree, cumulative probability chart and risk profile chart. The included TruSens add in allows you to realize sensitivity analisisis lik tornado, spider, surface, 3d column another reports. A little addin feature is the highlighting of best and optimal decision. But a disadvantage is the creation of an extra spreadsheet for each tree.

Figure 3. Tree plan model

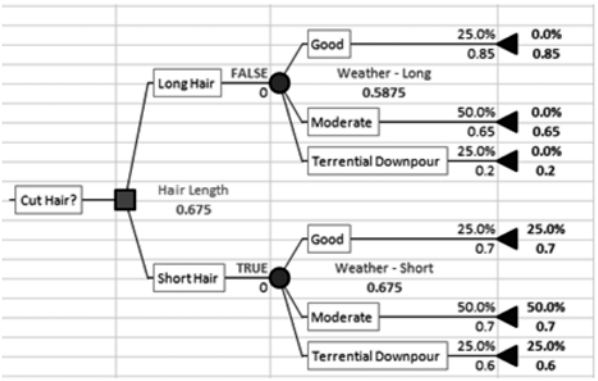
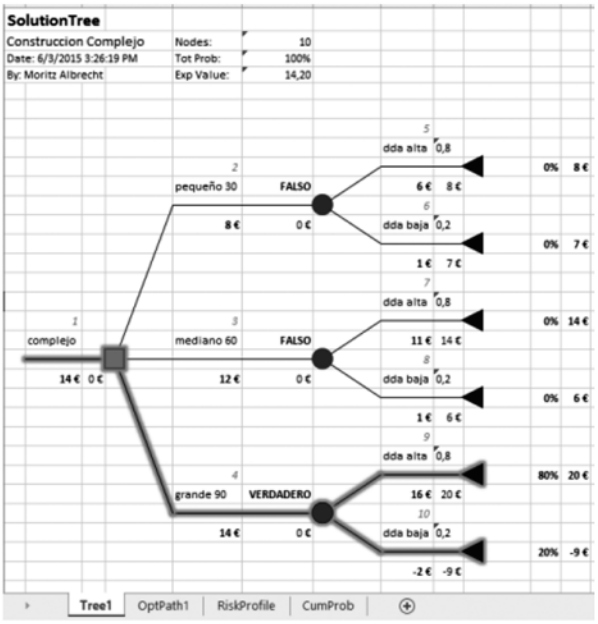


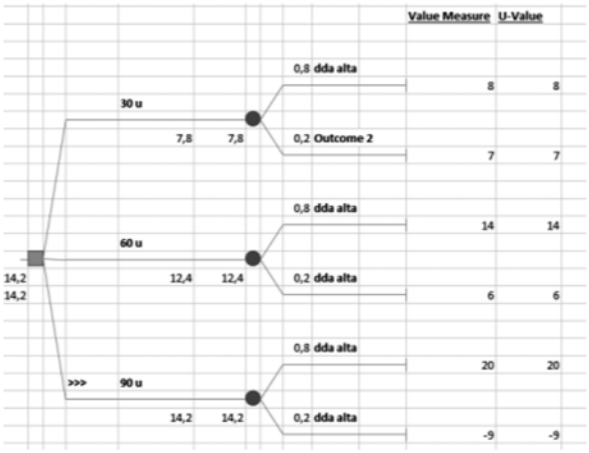
Figure 4. Tree plan solution



C. Simple Decision

Simple Decision is the only free add-in in this article which by that allows the user to edit the VBA macros. But this on the other hand means the restriction to only basic features like drawing, not copying nor collapsing, and calculation of expected and rollback values. But a positive feature is the easy to change utility function in the ribbon bar. It does not include sensitivity nor risk analysis.

Figure 5. Tree plan model in simple decision



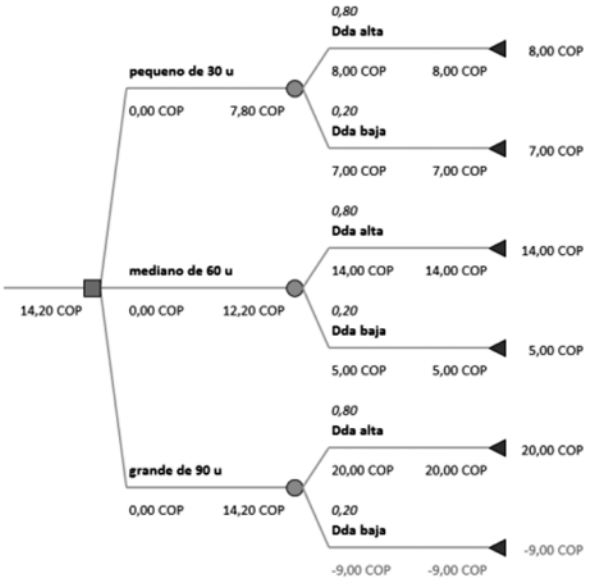
D. TreePlan

TreePlan is a good alternative for easily creating decisiontrees via evoked dialogue windows through Ctrl+Shift+T. It is possible to copy branches and edit Cells Objects and Columns easily. Optional addins allow MonteCarlo Simulation and Sensitivity Analysis. As well you are limited to one decision tree per spreadsheet.

E. Lumenaut Now

Lumenaut Now is a decisiontree add-in only available for older Excel versions, making it the only pro fact regarding its discontinued service.

Figure 6. Tree plan model in lumenaut now



IV. Standalone software

A. DTREG

DTREG is a predictive modelling software it is pronounced D-T-Reg and builds classification and regression decision trees, neural networks,support vector machine (SVM), GMDH polynomial networks, gene expression programs, K-Means clustering, discriminant analysis and logistic regression models that describe data relationships and can be used to predict values for future observations . DTREG also has full support for time series analysis. DTREG accepts a dataset containing of number of rows with a column for each variable. One of the variables is the "target variable" whose value is to be modeled and predicted as a function of the "predictor variables". DTREG analyzes the data and generates a model showing how best to predict the values of the target variable based on values of the predictor variables. DTREG can create classical, single-tree models and also TreeBoost and Decision Tree Forest models consisting of Ensembles of many trees. DTREG also can generate Neural

Networks, Support Vector Machine (SVM), Gene Expression Programming/Symbolic Regression, K-Means clustering, GMDH polynomial networks, Discriminate Analysis, Linear Regression, and Logistic Regression models. DTREG includes a full Data Transformation Language (DTL) for transforming variables, creating new variables and selecting which rows to analyze.

B. TreeAge

TreeAge is a software to model and analyse complex decisions.

Referencias

[1] Schawel, C., & Billing, F. (2014). *Topp 100 Management Tools.* Wiesbaden: Springer Fachmedien.

[2] weebly. (n.d.). *A Simple Decision Tree Problem*. Retrieved 06 10, 2015, from http://ams-decisiontreeanalysis.weebly.com/how-to-solve-problems.html

[3] Dahan, H., S., C., Rokah, L., & Maimon, O. (2014). Proactive Data Mining with Decision Trees. Springer.

Figure 7. Tree age interface

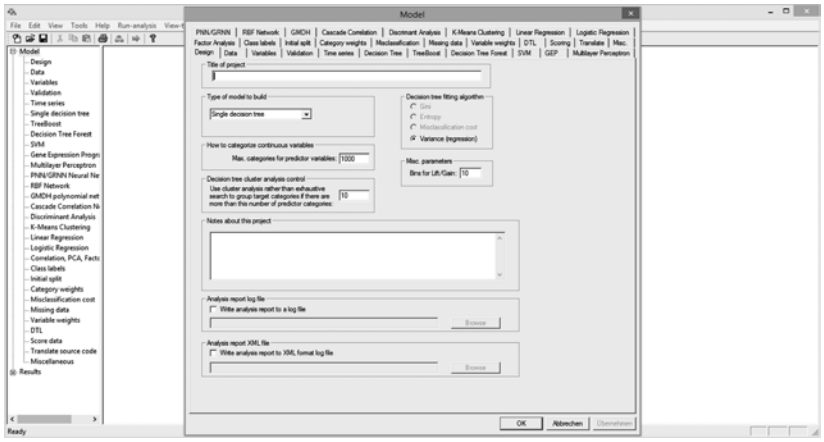


Figure 8. Tree age model