Newton Trees

F. Martínez, V. Estruch, C. Ferri, J. Hernández-Orallo and M.J. Ramírez-Quintana

Departmento de Sistemas Informáticos y Computación (DSIC)
Technical University of Valencia (UPV)
Valencia, Spain

23rd Australasian Joint Conference on Artificial Intelligence

December 7-10, 2010
Adelaide, South Australia

Outline

1. Introduction
   - Decision Tree Learning
   - Antecedents
   - The solution proposed
2. Newton Trees
   - Newton Tree Generation
   - Using Newton Trees
3. Experimental Results
4. Conclusions and Future Work

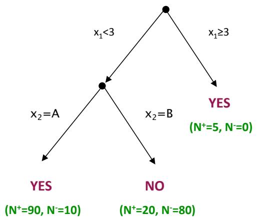
Motivation

Decision Tree Learning

One of the most popular techniques for classification

- *divide – and – conquer* covering of the problem space
- use of decisions defined over univariate conditions
  - internal nodes: test on the values of a selected attribute
  - leaves: labelled with a class label

Example



The problem

Advantages of Decision Trees

- Comprehensibility of the models (set of If-Then rules).
- They work remarkably well which has facilitated their wide use.

Disadvantages of Decision Trees

- Decisions are crisp (although leaves are generally not pure).
- Sometimes, a ranking is needed.
- They can only deal with nominal and numerical data.
- Different condition expressions depending on the attribute data type.

  Example: $x_1 < 3$, $x_1 \geq 3$
  $x_2 = A$, $x_2 = B$

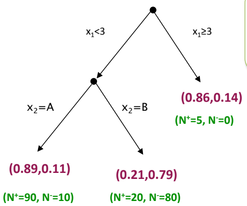
Probability Estimation Trees, PET's

[Ferri et al. 2003], [Provost & Domingos 2003]

The output is a probability rather than a crisp decision.

- *Divide – and – conquer* philosophy for constructing the tree and for making a prediction.
- They have been successfully used whenever the goal is to produce good rankings or good probability estimations.

Example



Laplace correction

$$p(C) = \frac{n_C + 1}{n + 2}$$

Centre Splitting [Thornton 2000],

A machine learning method which consists in dividing the input space in different regions where each region is represented by a centre.

- For each iteration,
  - a centre is calculated for every different class which is presented in the area,
  - every example is associated to its nearest centre.
- This process is repeated until the area is pure.

One of the special features of this method is that the examples are managed as a whole.

Distance-based Decision Trees [Estruch 2008],

A learning strategy that joins centre splitting and decision tree learning techniques.

- Partitions are made only taking into account one attribute at a time which is selected depending on a heuristic function (Gain Ratio, GINI index, etc.) (classical decision trees)
- Partitions only consider the distances and class distribution of the selected attribute (centre splitting)
  - any kind of data type can be handled.
  - All data types are handled in a uniform way.
- Centroids are replaced by prototypes to handle any datatype. Example: the centre of the set of lists [a, a], [a, a, a], [a, a, a, a, a] is a list containing 3.33 a's.

A new Stochastic Distance-based Probability Estimation Tree Learning Technique, Newton Trees

- Based on the principle of attraction, which is used
  - to construct the tree
  - to derive the probabilities
  - to use and represent the tree
- It uses the notions of distance (in a univariate way) and prototypes.
- Stochastic in the way in that the tree is constructed and used.
- We provide a graphical representation of the trees to easily interpret them.

Newton Tree Generation

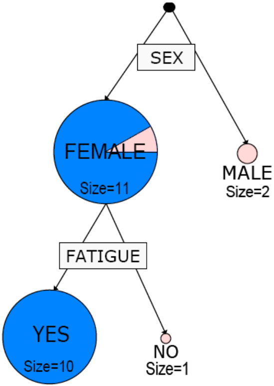
For each attribute x_r and for each class i,

1. a prototype $\pi_{r,i}$ is calculated as the attribute value with the lowest mean distance to the elements of the class.
2. the splitting attribute is selected according to one of the well-known splitting criteria (Gain ratio).
3. the split proceeds by associating every instance to its closest attribute prototype according to the following attraction function

$$\boxed{attraction(e, \pi) = \frac{m_e m_\pi}{d(e,\pi)^2} = \frac{m_\pi}{d^2}}$$

4. if the partition is impure then go to 1 else exit.

Newton Tree Representation: an Example



$$d_{nom}(x,y) = \begin{cases} 1 & if \quad x \neq y \\ 0 & if \quad x = y \end{cases}$$

Probability Calculation

Given a new example e and a Newton tree T, the objective is to calculate the probability vector at the root of T, $\overrightarrow{p}(root, e)$.

STEP 1: Compute the probability $\hat{p}(\nu, e)$ that e falls into node ν (coming from its parents):

$$\hat{p}(\nu, e) = \frac{attraction(e,\nu)}{\sum_{\mu \in Children(Parent(\nu))} attraction(e,\mu)}$$

Probability Calculation

STEP 2: Estimate the probability vector of example e at node ν, $\overrightarrow{p}(\nu, e) = \langle p_1(\nu, e), \ldots, p_c(\nu, e)\rangle$, where $p_i(\nu, e)$ denotes the probability that e belongs to class i at node ν.

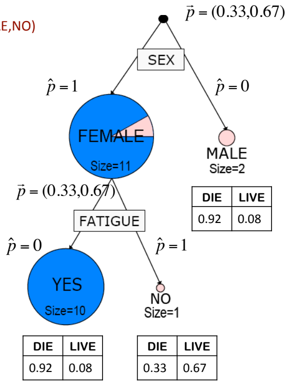
$$\forall \nu \in T : \overrightarrow{p}(\nu, e) = \begin{cases} \sum_{\mu \in Children(\nu)} \hat{p}(\mu, e) \cdot \overrightarrow{p}(\mu, e) & \text{if } \nu \neq \text{leaf} \\ \langle Laplace(1, \nu), \ldots, Laplace(c, \nu)\rangle & \text{if } \nu = \text{leaf} \end{cases}$$

where $Laplace(j, \nu)$ is the Laplace correction of the frequency of

elements of class j in leaf ν.

Example

(FEMALE,NO)



Experimental Setting

The aim is to compare Newton Trees with a well-known Probability Estimation Tree

- unpruned J48 (implemented in Weka) with Laplace smoothing in the leaves.
- Splitting criterion: Gain Ratio.
- 30 datasets from the UCI repository (instances with missing values were removed).
- 20 x 5-fold cross validation (100 learning runs for each pair of dataset and method).
- Evaluation metrics: accuracy, as a qualitative measure of error, AUC (Area Under the Curve) as a measure of ranking quality, and MSE (Mean Squared Error) as a measure of calibration and refinement quality.

Experimental Results

Average results

	Newton Trees			Unpruned Laplace J48		
	Acc.	AUC	MSE	Acc.	AUC	MSE
Mean (All)	82,0907	**0,8664**	0,1004	80,7283	0,8419	0,1101
Mean ($c = 2$)	83,6503	**0,8665**	0,1146	81,3388	0,8310	0,1334
Mean ($c > 2$)	80,3084	**0,8662**	0,0843	80,0307	0,8544	0,0834
Mean (Nominal)	90,1592	**0,9311**	0,0688	87,3099	0,8944	0,0803
Mean (Numerical)	79,7034	**0,8599**	0,1175	79,4223	0,8482	0,1265
Mean (Mixed)	77,2053	**0,8102**	0,1093	75,8881	0,7811	0,1179

Wilcoxon signed-ranks test with a confidence level of $\alpha = 0.05$

	Acc.	AUC	MSE
All	14/6/10	18/8/4	14/4/12
Nominal	2/3/4	5/2/2	2/0/7
Numerical	5/3/4	5/5/2	7/3/2
Mixed	7/0/2	9/0/0	5/1/3

Conclusions

- We have defined a novel probability estimation tree learning method which is based on computing prototypes and applying an Inverse Square Law in order to derive an attraction force which is then converted into a probability.
- The use of prototypes allows us for the use of our trees for any kind of datatype.
- The number of different splits to evaluate at each node is equal to the number of attributes and does not depend on midpoints or the size of the dataset.

Future Work

- We are adapting Newton Trees to deal with missing values in both the training set and the test set.
- To study other splitting criteria, like AUC.
- To apply/extend Newton Tree philosophy to other machine learning tasks (regression, clustering).

Thanks for your attention