An improved Id3 algorithm for clinical data classification

-Data Mining

119003228 N.L.Veerendra
119003080 K.Gangadhar
119003107 P.Rohith Surya

Smt. Kamakshi.S

Paper:

Yang, Shuo, Jing-Zhi Guo, and Jun-Wei Jin. "An improved Id3 algorithm for medical data classification." *Computers & Electrical Engineering* 65 (2018): 474-487.

Url:

https://www.sciencedirect.com/science/article/pii/S00457906 1732517X

Objective

- To improve the Id3 algorithm overcomes multi-value bias problem when selecting test/split attributes and to solve the issue of numeric attribute discretization.

Introduction:

- Improved Id3 algorithm for disease prediction
- Features:
  - balance function for test attribute selection.
  - numeric attribute discretization strategy.
  - new rule-based heuristic method for classified representation.

Basic principles of Id3

- The Id3 algorithm
  - chooses test attributes by calculating and comparing their information gains.
  - Let S_i be the number of samples in class C_i. The expected information amount required to classify S is given by :

$$I(S_1, S_2, \ldots\ldots, S_m) = -\sum_{i=1}^{m} p_i log_2 p_i$$

  - p_i probability of instances in S belonging to class C_i.

Basic principles of Id3

- The required information amount (i.e.,entropy) of attribute A to split the training dataset S is given by:

- $E(A) = \sum_{j=1}^{v}(\frac{(s_{1_j}+s_{2_j}+\cdots+smj)}{s} \times I(s_{1j}, s_{2j}, \ldots\ldots, smj))$
- The less information amount required, the more purity of a sub-dataset is.
- The information gain of *A* is defined as:

$$\text{InfoGain}(A) = I(S_1, S_2, \ldots, S_m) - E(A)$$

Problems of classical Id3 algorithm

- Bias on multi-variated attributes
  - The more different attribute values, the larger information gains.
- Helplessness with numeric attributes
  - Clasical ID3 algorithm works only on nominal data.
  - if the default filter of NumericToNominal in WEKA is used to discretize the attributes, the performance of Id3 algorithm degrades dramatically.

Problems of classical Id3 algorithm

- Loosing relations among attributes
  - uses a tree structure to represent a classifier model which is hard to interpret and understand.
  - the tree representation is neither an optimal method for storage nor a good way for interpretation.(takes more memory and uses recursive calls to traverse the tree)

Improvements on ID3

- Balance Function
- Discretization of numeric attributes
- Rule representation of classifier model

Balance Function

- Solves the problem of Bias on multi-variated attributes
- The Balance Function is a monotonically increasing function of number of attribute values.
- the monotonic trend of a balance function *f* (*n*) should be comparatively flat in the range of *n* , which guarantees that the balance function is not over-balancing
- The following function satisfies all the constrains

$$f(n) = \left| log_2\left(\sqrt{\mathrm{n}+1}\right) \frac{\cos(3.5n - 1.5)}{1.8} \right|, (\mathrm{n} > 0)$$

- Gain is recomputed as

Gain (A) = InfoGain (A) / f (n)

Discretization of numeric attributes

- Suppose f (x) is a concave function on the interval I ,

$\forall x_1, x_2, \ldots, x_n \in I$, $\lambda_1, \lambda_2, \ldots, \lambda_n > 0$, $\lambda_1 + \lambda_2 + \ldots + \lambda_n = 1$, then

$$\lambda_1 f(x_1) + \ldots + \lambda_n f(x_n) \leq f(\lambda_1 x_1 + \ldots + \lambda_n x_n)$$

- Based on the above property,
- $I(S_1, S_2, \ldots, Sm) = -\sum_{i=1}^{m} p_i log_2 p_i \leq -log_2 \sum_{i=1}^{m} {p_i}^2$

Discretization of numeric attributes

- In a real application, if $p_i - p_j = p \rightarrow 0$, then
- $I(S1, S2, \ldots., Sm) = -\sum_{i=1}^{m} p_i log_2 p_i \cong -log_2 \sum_{i=1}^{m} {p_i}^2$
- Enlarging the number of bins for splitting datasets gives a small entropy
- We first create one interval for each label in a class attribute in the training dataset.
- Choose minimum and maximum values of the numeric attribute (e.g.,A) in a sub-dataset (e.g.,s_1) as two endpoints of the interval.

Algorithm to discretize a numeric attributes

Discretize(Ins,Att,k):Ins-training dataset; Att – a numeric attribute; k-the number of different labels in a class attribute.

Input: Att,Ins,k
Output: Intervals of Att

1) Ins[][] = Partition(Ins,k);//function Partition divides Ins in to k parts, stored in 2-dimensional array Inss.
2) Sort(Inss,Att);//function Sort sorts samples in all sub-datasets of Inss based on the value of Att in ascending order.
3) For i ∈ |Inss|//|Inss| is the number of sub-datasets in Inss.
4) Inters[i] = Generate interval [Min(Inss[i],Att),Max(Inss[i],Att)] for each sub-dataset Inss[i];//Min(Inss(i)) is the minimum value of Att in Inss[i];Max(Inss[i]) is the maximum value of Att in Inss[i];Inters[i] is the corresponding interval of Inss[i].
5) End_for

Algorithm to discretize a numeric attributes

6) For i ∈ |Inters|//|Inters| is the number of intervals
7) subInts[i][] = Split(Inters[i]);//function Split divides each Inters[i] into independent sub-intervals based on numerical continuity and stores them in array subInts[i][].
8) End_for
9) For I ∈ |subInts|//|subInts| is the number of total sub-intervals.
10) Merge(subInts);//function Merge combines adjacent sub-intervals based on the three basic rules.
11) End_for

Improved Id3 algorithm

Id3_improved algorithm: Generate_Decision_tree(*S* , attribute list). *S* is a training dataset.

Input: training dataset; numeric class attribute is discretized before constructing a classifier model.
Output: a rule-based classifier model.

1) Create a node M
2) If all instances in a sub-dataset are in the same class (e.g. C), execute (3)
3) Return M as a leaf node and mark it with class C
4) If attribute list is empty, then execute (5)
5) Return M as a leaf node and mark it with the majority class in the dataset
6) If attribute list is not empty, then select an attribute with the greatest InfoGain/f(n) from the attribute list as a test attribute
7) Mark the node M as the test attribute
8) (8) If it is a numeric attribute, discretize it based on the class attribute values; Then execute (9)

Improved Id3 algorithm

9) For each value *a i* of the test attribute (e.g., A)
10) Regard *A* = *a i* as a testing condition and generate a corresponding branch from the node M
11) Let *S i* be the sub-dataset when *A* = *a i*
12) If *S i* is empty, then execute (13)
13) Add a leaf node and mark it with the majority class in the dataset
14) Else mark it with the return value of Generate_Decision_tree(*S i* , {new attribute list|{attribute list} −*A* })
15) Return rule-based decision list.

Rule representation of classifier model

- TransformTreeToRules approach generates rule-based classifier models
  - representing rules as “IF...THEN ... ”.
- Conjunction of all attribute-value pairs in the path constructs the IF part of the rule, while the class label in a leaf is the THEN part

Heuristic strategy

- Heuristic strategy of four steps for rule pruning are used to shrink the size of the rulebase:
  - **Step 1** : Supplement each rule with irrelevant values. Irrelevant values are the values of irrelevant attributes which can be deleted or replaced by any values from the same domain values without affecting the correctness of the rule.
  - **Step 2** : If a rule (*e.g* ., r_1) has *n* conditions in the head part and the former *i* conditions are different from other rules, then the latter ($n-i+1$) conditions can be deleted from r_1.
  - **Step 3** : All rules are assigned to different groups in a decedent order according to the number of same conditions in the head part.
  - **Step 4** : The set of rules is further reduced based on the MDL principle finding the shortest description of rules of the same group by dropping repeated conditions in their head parts.

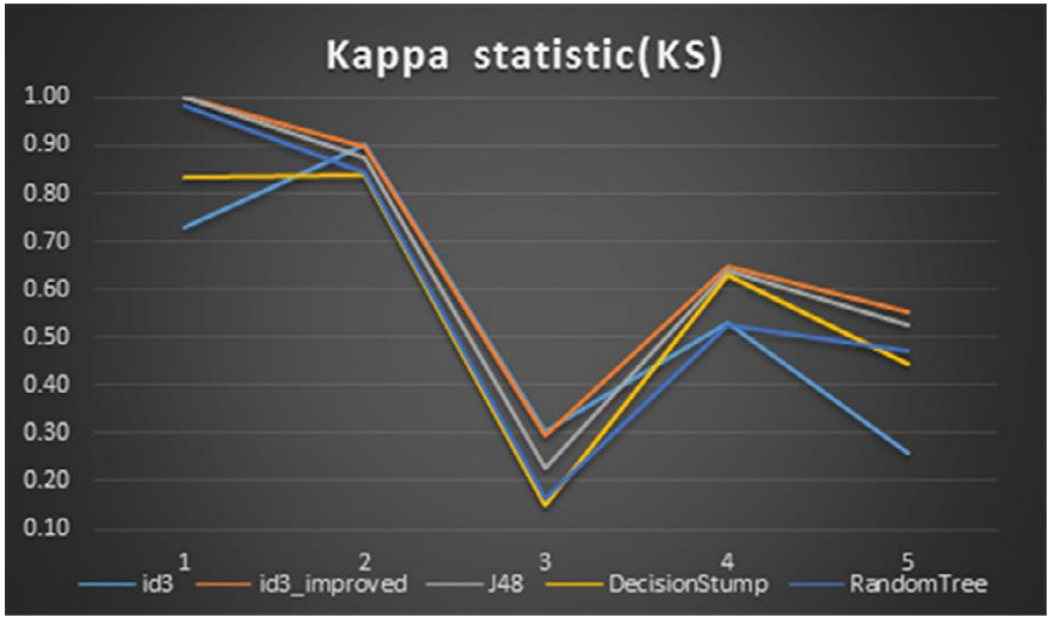
Data sets(from UCI)

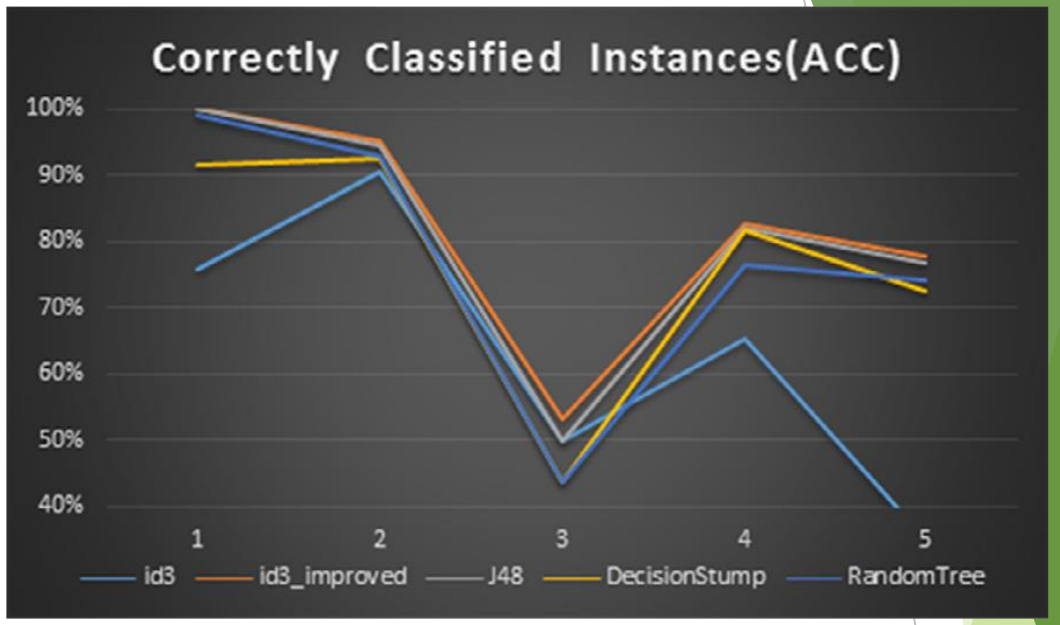
Experiment datasets.

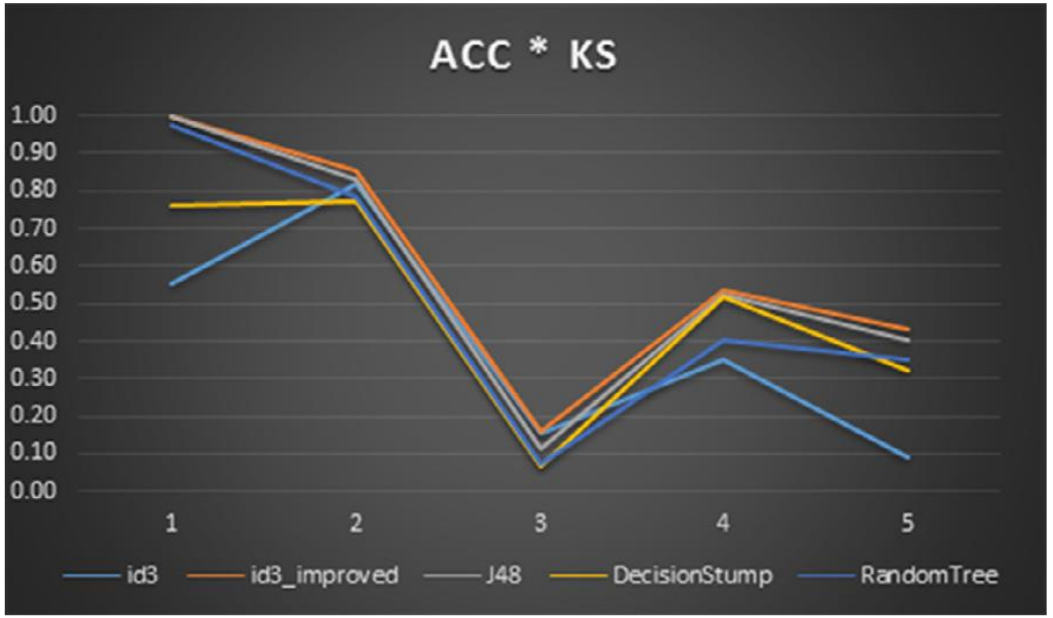
Name of dataset	No. of samples	No. of attributes	No. of classes
Acute inflammations	120	6	2
Wisconsin breast cancer	699	10	1
Lung cancer	32	56	2
Mammographic mass	961	6	1
Statlog (Heart)	270	13	1

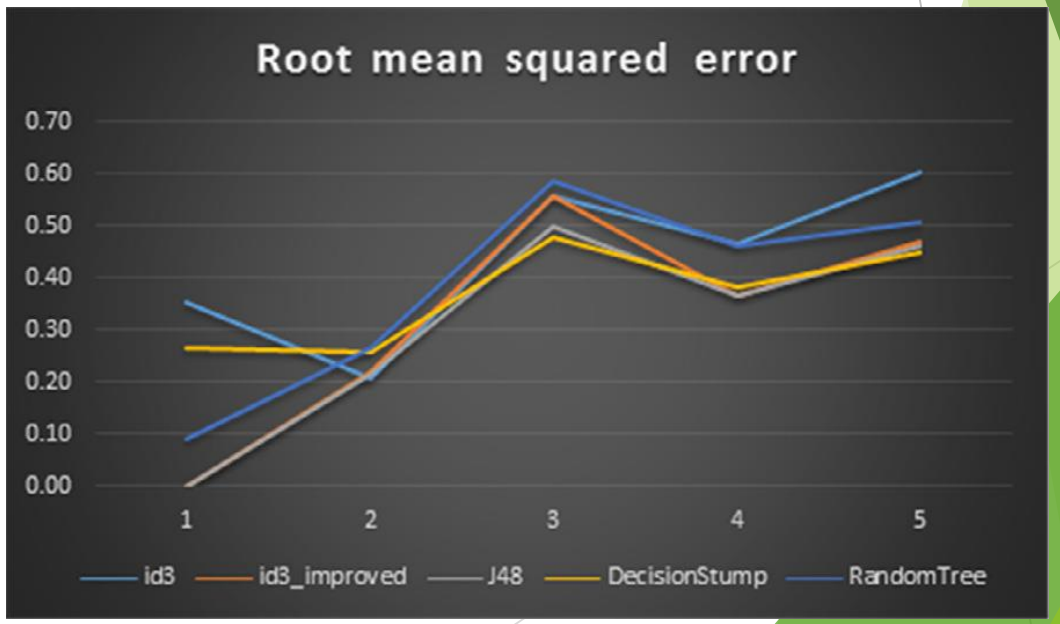
Data Sets used

- Acute inflammations data set: Diagnosing of the acute inflammation of the urinary bladder and acute hephritis. Instances: 120 attributes: 6
- Wisconsin breast cancer data set: Obtained from University of Wisconsin Hospitals. Instances: 699 attributes: 10
- Lung cancer data set: Instances: 32 attributes: 56
- Mammographic mass data set: The data set used to predict the severity (benign or malignant) of a mammographic mass lesion. Instances: 961 attributes: 6
- Statlog (Heart) data set: Instances: 270 attributes: 13


Kappa statistic(KS)
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
1
2
3
4
5
id3
id3_improved
J48
DecisionStump
RandomTree


Correctly Classified Instances(ACC)
100%
90%
80%
70%
60%
50%
40%
1
2
3
4
5
id3
id3_improved
J48
DecisionStump
RandomTree


ACC * KS
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
1
2
3
4
5
id3
id3_improved
J48
DecisionStump
RandomTree


Root mean squared error
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
1
2
3
4
5
id3
id3_improved
J48
DecisionStump
RandomTree

Comparison in memory saving