Practical Data Mining with the WEKA tool

CO832 - Data Mining and Knowledge Discovery
University of Kent

Jurgen Palsma 28 February 2018 jurgen.palsma@gmail.com

1. Analysis of the effect of different J48 parameter settings in the predictive accuracy of the constructed decision trees

From all the parameters provided by WEKA that allows us to tune and personalise a J48 algorithm, the ones which affect its predictive accuracy are those which globally affect the algorithm's pruning mechanisms. WEKA gives us a high-order choice of two pruning methods; the default C4.5 technique, and a reduced error pruning technique (-B, REP). Both of these techniques use subtree raising (SRA) by default, but can use a subtree replacement technique (-S, SRE), which give the user a total panel of 4 different pruning technique combinations.

When using REP with either SRA or SRE, we can affect accuracy by tuning the proportion of data used for pruning by changing the -N parameter. The more we increase it, the smaller the proportion of data will be used for pruning, and vice versa. If the proportion is too small, we will tend to overfit the data because of the lack of pruning (*Fig. 3, run 3)* , which causes a reduction of the PA, and we will in turn over-generalise if the proportion is too high because of the intensive pruning (*Fig. 3, run 7)* .

When using standard C4.5 pruning with SRA or SRE, we have a confidence factor parameter at our disposal (-C, CF). When we lower our confidence factor, the error estimate (a value representing the worst case scenario in which our node will misclassify most data reaching it) of our nodes will increase and will increase the probability of that node to be replaced by another node or tree, depending on your pruning method. Thus, if CF is too low, our tree will be highly pruned, and thus too general and PA will drop (*Fig. 4)*. Parameters that affect our J48s' PAs regardless of their pruning method include the minimum instances of a leaf (-M), which influence the generality of our trees by constraining our algorithm to more general leaves if we increase -M or by allowing it more varied leaves by decreasing -M.

2. Analysis of the effect of different J48 parameter settings in the comprehensibility of the constructed decision trees

Even if evaluating the comprehensibility of a tree is largely subjective, it can be agreed that a smaller tree is generally more comprehensible: as a result, parameters which influence pruning and generalisation as seen in the first part of this report such as confidence and minimum instances of a leaf can be used to produce smaller, more general and more comprehensible trees. Unfortunately, generalising a tree can affect PA, and thus one has to compromise between PA and comprehensibility. For example, in *Fig. 2*, the model is very comprehensible as it is only 17 nodes big, using 6 different attributes, and in sequences that seem logical (for example, the attribute "bore" and "stroke" are often coupled in a metric used in vehicle safety and power evaluations, called engine displacement[1]), but only scores a low 65.8% of PA because of its over-generalising size and nodes.

In addition, one can choose to split nodes which condition nominal attribute in a binary way (-B), which I find adds a lot of comprehensibility of our data as some models have more than 20 child nodes on a single node (see *Fig. 1*) when conditioning on the "make" attribute. Not only is this visually painful but seems also of lesser value when classifying cars; one "make" can produce cars which highly vary in "safety" (the attribute we are looking to classify our individuals on). In *Fig. 6* you can see that by using the -B parameter we still have a classification that is heavily dependant on the "make" attribute but where most "makes" are still divided in different sub-trees, which cancels overgeneralization on that attribute and permits analysis of different "makes". Finally, by analysing the raw data set, we see that the "makes": "volvo" and "volkswagen" have significantly uniform "symboling" values: this can be seen in *Fig. 7*, which I choose as the best model since it is both comprehensible as it discerns the special cases of "volvo" and "volkswagen", taking in consideration engine displacement[1] and vehicle size and weight, whilst remaining acceptably accurate (PA just under 70%, see Fig. 5, run 3).

Another small contributor to comprehensibility is the -L parameter, which does not affect the model itself but allows you to have further visualisation capacities when visualising your tree in a graphical mode.

1. "Engine Displacement." *Wikipedia*, Wikimedia Foundation, 28 Feb. 2018, en.wikipedia.org/wiki/Engine_displacement.

Appendix

```
num-of-doors = four
|  wheel-base <= 101.2
|  |  height <= 51.6: 1 (10.56/1.0)
|  |  height > 51.6
|  |  |  curb-weight <= 2050
|  |  |  |  horsepower <= 69: 1 (7.0)
|  |  |  |  horsepower > 69: 0 (4.0)
|  |  |  curb-weight > 2050
|  |  |  |  height <= 55.6
|  |  |  |  |  price <= 13860: 0 (31.56)
|  |  |  |  |  price > 13860
|  |  |  |  |  |  normalized-losses <= 168: 2 (2.0)
|  |  |  |  |  |  normalized-losses > 168: 0 (2.0)
|  |  |  |  height > 55.6
|  |  |  |  |  body-style = hardtop: 2 (0.0)
|  |  |  |  |  body-style = wagon: 0 (4.0)
|  |  |  |  |  body-style = sedan: 2 (8.0)
|  |  |  |  |  body-style = hatchback: 2 (0.0)
|  |  |  |  |  body-style = convertible: 2 (0.0)
|  wheel-base > 101.2
|  |  make = alfa-romero: -1 (0.0)
|  |  make = audi: 1 (3.0)
|  |  make = bmw: 0 (3.0/1.0)
|  |  make = chevrolet: -1 (0.0)
|  |  make = dodge: -1 (1.0)
|  |  make = honda: -1 (0.0)
|  |  make = isuzu: -1 (0.0)
|  |  make = jaguar: 0 (2.0)
|  |  make = mazda: 0 (2.0)
|  |  make = mercedes-benz: -1 (5.0/1.0)
|  |  make = mercury: -1 (0.0)
|  |  make = mitsubishi: -1 (0.0)
|  |  make = nissan: -1 (0.0)
|  |  make = peugot: 0 (11.0)
|  |  make = plymouth: -1 (1.0)
|  |  make = porsche: -1 (0.0)
|  |  make = renault: -1 (0.0)
|  |  make = saab: -1 (0.0)
|  |  make = subaru: -1 (0.0)
|  |  make = toyota: -1 (7.0)
|  |  make = volkswagen: -1 (0.0)
|  |  make = volvo
|  |  |  normalized-losses <= 98: -1 (8.0)
|  |  |  normalized-losses > 98: -2 (3.0)
num-of-doors = two
|  city-mpg <= 22
|  |  wheel-base <= 99.1: 3 (23.0/2.0)
|  |  wheel-base > 99.1
|  |  |  engine-type = dohc: 3 (2.0)
|  |  |  engine-type = dohcv: 0 (0.0)
|  |  |  engine-type = l: 0 (0.0)
|  |  |  engine-type = ohc: 0 (6.0/2.0)
|  |  |  engine-type = ohcf: 0 (0.0)
|  |  |  engine-type = ohcv: 1 (3.0/1.0)
|  |  |  engine-type = rotor: 0 (0.0)
|  city-mpg > 22
|  |  bore <= 3.39
|  |  |  engine-size <= 103
|  |  |  |  width <= 64.2
|  |  |  |  |  wheel-base <= 89.5: 2 (3.0)
|  |  |  |  |  wheel-base > 89.5: 1 (26.44/1.0)
|  |  |  |  width > 64.2: 2 (4.0)
|  |  |  engine-size > 103
|  |  |  |  compression-ratio <= 8.5: 3 (4.0)
|  |  |  |  compression-ratio > 8.5
|  |  |  |  |  city-mpg <= 26: 1 (3.0)
|  |  |  |  |  city-mpg > 26: 0 (3.44/1.0)
|  |  bore > 3.39: 2 (12.0)
```

Figure 1: A textual representation of a J48 tree with default parameters (see Fig 5.1 for parameters).

run	Scheme & parameters	% correctly classified instances	size	leaves	comment
1	weka.classifiers.trees.J48 -R -N 3 -Q 1 -M 2	64.3902	42	32	default
2	weka.classifiers.trees.J48 -R -N 5 -Q 1 -M 2	71.7073	53	37	more folds = less pruning data = less overgeneralization = more accuracy but bigger trees
3	weka.classifiers.trees.J48 -L -R -N 2 -Q 2 -M 5	64.878	17	9	2 folds: 50% of data is used for pruning = very high pruning
4	weka.classifiers.trees.J48 -R -N 3 -Q 1 -M 4	66.3415	45	35	more items in leaves = less overfitting on small samples
5	weka.classifiers.trees.J48 -R -N 5 -Q 1 -B -M 2	68.2927	29	15	with binary splits, not very accurate but small comprehensible tree, and conditions on same attribute.
6	weka.classifiers.trees.J48 -L -R -N 2 -Q 2 -M 2	70.2439	87	74	low folds, huge tree, high PA but not optimal because of overfitting
7	weka.classifiers.trees.J48 -L -R -N 50 -Q 2 -M	57.0732	25	23	High folds cause very low PA and small tree

Figure 3: A table presenting the result of seven distinct runs of J48 with REP

run	scheme & parameters	% correctly classified instances	size	leaves	comment
1	weka.classifiers.trees.J48 -C 0.01 -M 2	65.8537	40	22	min confidence
2	weka.classifiers.trees.J48 -C 0.03 -M 2	77.561	47	28	low confidence
3	weka.classifiers.trees.J48 -C 0.49 -M 2	82.439	69	49	high confidence
4	weka.classifiers.trees.J48 -C 0.51 -M 2	84.878	88	65	Very high confidence
5	weka.classifiers.trees.J48 -C 5.0 -M 2	84.878	88	65	Very high confidence

Figure 4: A table presenting the result of four distinct runs of J48 with varying confidence factor

run	scheme	% correctly classified instances	size	leaves
1	weka.classifiers.trees.J48 -C 0.25 -M 2	81.9512	69	49
2	weka.classifiers.trees.J48 -L -S -C 0.4 -M 12 -A	65.8537	17	9
3	weka.classifiers.trees.J48 -L -C 0.05 -B -M 8 -A	69.2683	23	12

Figure 5: A table presenting the result of three distinct runs of J48 with varying comprehensibility

```
make = volvo
|   normalized-losses <= 98.0: -1 (8.0)
|   normalized-losses > 98.0: -2 (3.0)
make != volvo
|   num-of-doors = four
|   |   height <= 51.6: 1 (10.54/1.0)
|   |   height > 51.6
|   |   |   make = saab: 2 (3.0)
|   |   |   make != saab
|   |   |   |   make = audi
|   |   |   |   |   wheel-base <= 102.7: 2 (2.0)
|   |   |   |   |   wheel-base > 102.7: 1 (3.0)
|   |   |   |   make != audi
|   |   |   |   |   make = volkswagen
|   |   |   |   |   |   wheel-base <= 98.8: 2 (5.0)
|   |   |   |   |   |   wheel-base > 98.8: 0 (3.0)
|   |   |   |   |   make != volkswagen
|   |   |   |   |   |   curb-weight <= 2050.0
|   |   |   |   |   |   |   horsepower <= 69.0: 1 (7.0)
|   |   |   |   |   |   |   horsepower > 69.0: 0 (4.0)
|   |   |   |   |   |   curb-weight > 2050.0
|   |   |   |   |   |   |   num-of-cylinders = five: -1 (3.0)
|   |   |   |   |   |   |   num-of-cylinders != five
|   |   |   |   |   |   |   |   normalized-losses <= 74.0: -1 (7.78/1.54)
|   |   |   |   |   |   |   |   normalized-losses > 74.0: 0 (55.76/4.63)
|   num-of-doors != four
|   |   city-mpg <= 22.0
|   |   |   body-style = sedan: 0 (4.0/1.0)
|   |   |   body-style != sedan
|   |   |   |   num-of-cylinders = five: 0 (2.0)
|   |   |   |   num-of-cylinders != five: 3 (28.0/5.0)
|   |   city-mpg > 22.0
|   |   |   bore <= 3.39
|   |   |   |   make = volkswagen
|   |   |   |   |   body-style = sedan: 2 (2.0)
|   |   |   |   |   body-style != sedan: 3 (2.0)
|   |   |   |   make != volkswagen
|   |   |   |   |   height <= 49.7: 3 (2.0)
|   |   |   |   |   height > 49.7
|   |   |   |   |   |   make = mitsubishi: 2 (4.0/1.0)
|   |   |   |   |   |   make != mitsubishi
|   |   |   |   |   |   |   wheel-base <= 89.5: 2 (3.0)
|   |   |   |   |   |   |   wheel-base > 89.5: 1 (30.93/3.46)
|   |   |   bore > 3.39: 2 (12.0)
```

Figure 6: A textual representation of a J48 model where conditions on nominal attribute split in a binary way.

```
make = volvo: -1 (11.0/3.0)
make != volvo
|   num-of-doors = four
|   |   height <= 51.6: 1 (10.54/1.0)
|   |   height > 51.6
|   |   |   make = volkswagen: 2 (8.0/3.0)
|   |   |   make != volkswagen
|   |   |   |   curb-weight <= 2050.0: 1 (11.0/4.0)
|   |   |   |   curb-weight > 2050.0
|   |   |   |   |   wheel-base <= 101.2
|   |   |   |   |   |   peak-rpm <= 5200.0: 0 (30.3)
|   |   |   |   |   |   peak-rpm > 5200.0: 2 (9.23/4.23)
|   |   |   |   |   wheel-base > 101.2
|   |   |   |   |   |   bore <= 3.39: -1 (13.0/4.0)
|   |   |   |   |   |   bore > 3.39: 0 (22.0/4.0)
|   num-of-doors != four
|   |   city-mpg <= 22.0
|   |   |   wheel-base <= 99.1: 3 (23.0/2.0)
|   |   |   wheel-base > 99.1: 0 (11.0/6.0)
|   |   city-mpg > 22.0
|   |   |   bore <= 3.39: 1 (43.93/15.46)
|   |   |   bore > 3.39: 2 (12.0)
```

Figure 7: A smaller model where conditions are also split on nominal attributes.