Experimenting with Weka Tree + Rule Classifiers Lab 2 - SEng 474 / CSC 578D Data Mining

Yudi Santoso

Credit: Cheng Chen, Maryam Shoaran

Now that you've learned how to use Weka-Explorer (Lab 1), let us go deeper and try to understand how some algorithms and methods work.

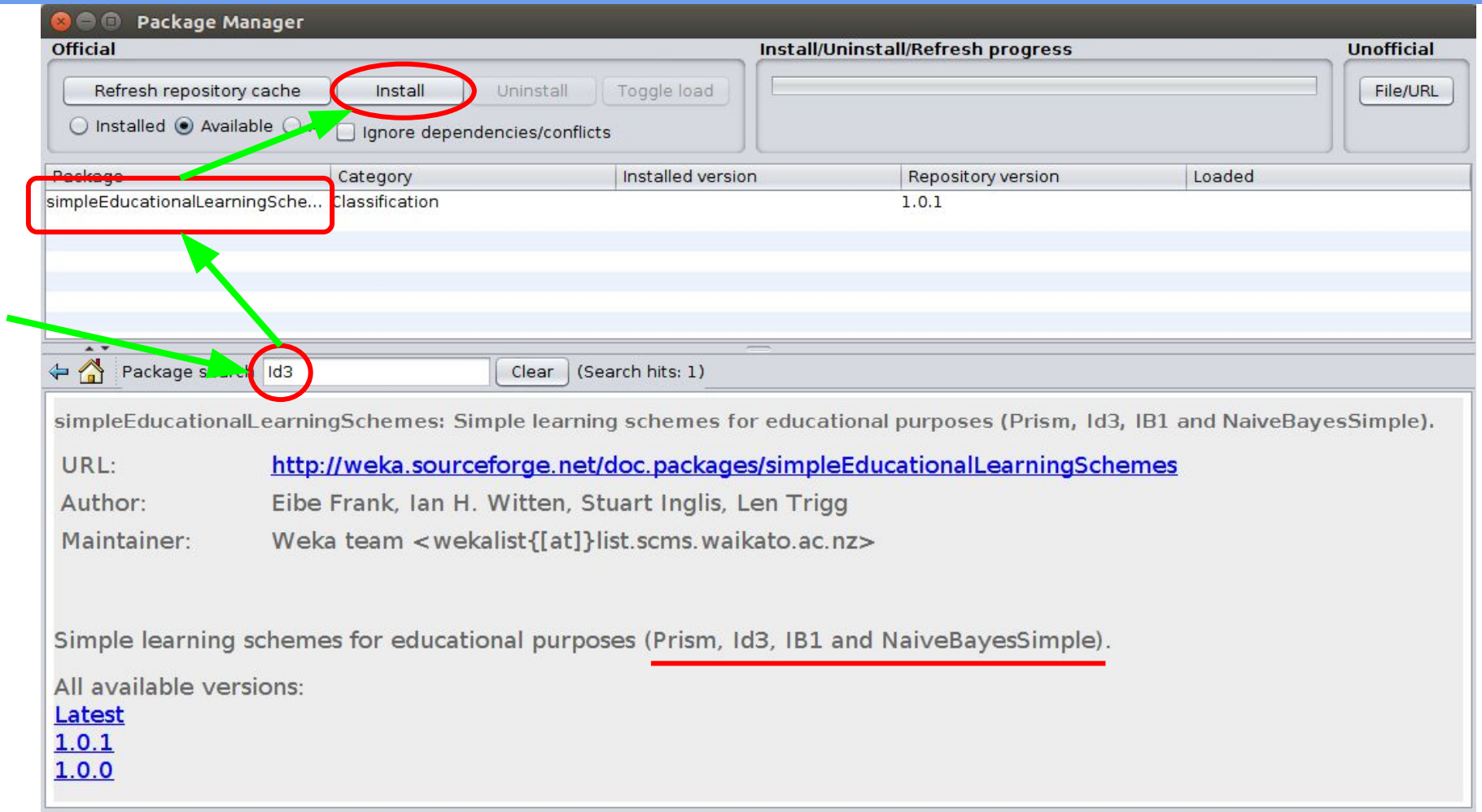
Weka Package Manager

Not all classes are installed by default. If you cannot find a particular classifier, you can search it in the Package Manager. For example: Id3.

- Close all windows except the Weka GUI Chooser window.
- Click `Tools > Package manager`
- Search for `Id3`, choose the package containing `Id3` (`simpleEducationLearningSchemes` which contains `Prism`, `Id3`, `IB1`, and `NaiveBayesSimple`)
- Click **Install**, then verify that the installation is successful
- Open Weka Explorer ➔ open a dataset ➔ `Classify` tab ➔ `Choose`
- The `Id3` classifier should now be listed under `trees`

Tools
> Package manager

Weka Package Manager



Id3 vs C4.5

- `C4.5` is an extension of `Id3`, where it can deal with
  - Missing values
  - Continuous attribute value ranges
  - Pruning
  - Rule derivation
  - etc
- There are also differences on how they use the decision tree.

Id3 C4.5 in Weka

Let's see some comparison in Weka-Explorer.

- Open `weather.nominal`. Classify with default options:
  - using `Id3` classifier
  - using `J48` classifier
- Compare the results (model, accuracy).
- `J48` has more parameters. Try different parameter values:
  - `Unpruned = True`
  - `minNumObj = 1`, etc
- Try another dataset: `contact-lenses`

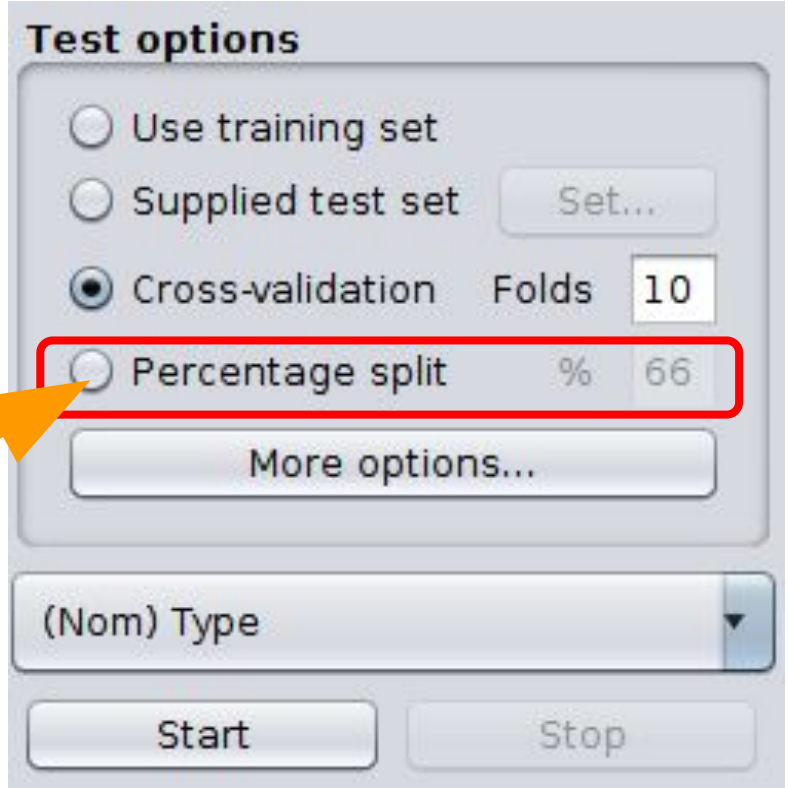
Random-Forest

- `RandomForest` is another tree classifier.
  - From a dataset of *M* attributes and *N* instances, it constructs a forest of random trees, each of $m < M$ attributes, and $n < N$ instances.
- Let's try it with Weka-Explorer-`Classify`.
  - Choose `tree` ➔ `RandomForest`
  - What parameters are available?
- Compare the `RandomForest` classifier output with `Id3` and `J48` (e.g., on `contact-lenses` dataset).

Classify Test Options

So far we have tried only the default cross-validation option.

Let us now try the percentage split option.



Training Percentage

- Open `contact-lenses` dataset.
- Choose `J48` tree classifier.
- Choose **Test Option** `Percentage split`. Specify percentage: 60%.
- Press **Start** to run the analysis.
- Record the accuracy.
- Repeat with percentage 0%, 20%, 40%, 80% and 100%.
- Sketch your observation as plot of accuracy as a function of percentage.
- What do you see? Can you explain?

Rule Classifiers

Decision tree is not the only way of doing classification. Rule classifiers, such as `ZeroR`, `OneR`, `Prism`, and `JRip`, build rules based on the training set.

- Try them out on some datasets (your choice). Use the default Cross-validation option.
- Study the output models.
- Compare the results among them, and to the results of `J48` and `RandomForest`.

```
Prism rules
----------
If astigmatism = no
   and tear-prod-rate = normal
   and spectacle-prescrip = hypermetrope then soft
If astigmatism = no
   and tear-prod-rate = normal
   and age = young then soft
If age = pre-presbyopic
   and astigmatism = no
   and tear-prod-rate = normal then soft
If astigmatism = yes
   and tear-prod-rate = normal
   and spectacle-prescrip = myope then hard
If age = young
   and astigmatism = yes
   and tear-prod-rate = normal then hard
If tear-prod-rate = reduced then none
```

....

Prism Rules

Baseline Analysis

`ZeroR` is the default classifier when we first open the Classify tab. Do you know why?

Let's look closer. Try it on weather.nominal dataset. What is the accuracy that you get?

Now, go back to the Preprocess tab. See that there are 14 instances, 9 with `play=yes`, and 5 with `play=no`. The majority is `play=yes`. What is 9/14?

Confirm your finding by looking at the description and trying out on other datasets.

Using Filters (1)

- `Id3` cannot deal with numerical attributes. Confirm this with `weather.numeric` dataset, for example.
- However, we can still proceed with this algorithm by preprocessing the data using a filter.
- `Preprocess` tab ➔ **Filter - Choose** ➔ `filters` ➔ `unsupervised` ➔ `attribute` ➔ `NumericToNominal`, ➡ **Apply**

Using Filters (2)

- Check under the `Classify` tab, `Id3` classifier is now enabled.
- Under the `Preprocess` tab, examine the values of the attributes after applying the filter.
- The effect of the filter can be undone by clicking **Undo**.
- There are many other filters available. Explore, and find out their functions.

Using Filters (3)

- Another filter is for dealing with missing values.
- For example:
  - Open `labor` dataset.
  - Click **Edit** - we can see that some attribute values are missing.
- **Filter - Choose** ➔ `filters` ➔ `unsupervised` ➔ `attribute` ➔ `ReplaceMissingValues`, ➡ **Apply**
- Find out how this method works.

Remove Attributes

Sometimes we get a dataset with some attributes that are not meant to be used for training, e.g., instance-ID. Also, sometimes we want to exclude some attributes to test the effect on the analysis.

- How to remove attributes:
  - `Preprocess` tab, select the attributes (check-box), then click **Remove**.
- Example:
  - Open `supermarket` dataset. Remove the departments attributes.

Closing

In this lab, we have learned some features in Weka, and some new learning algorithms. However, it is just a small part of the Data Mining field. Much more to be learned. So, keep exploring, and carry on!