ID3 USER MANUAL
version 0.01
June 19, 2009

Introduction

ID3A is an application that attempts to facilitate better usage of the ID3 algorithm as implemented in Sergio Fieren's ai4r rubygem.

But, what is ID3 exactly anyway? It is an algorithm that uses the idea of Information Entropy as described by this formula:

$$I_E(i) = -\sum_{j=1}^{m} f(i,j) \log_2 f(i,j).$$

While the formula looks intimidating to someone with little or now mathematical ability it is in essence Occam's Razor. To quote wikipedia "*When multiple competing hypotheses are equal in other respects, the principle recommends selecting the hypothesis that introduces the fewest assumptions and postulates the fewest entities. It is in this sense that Occam's razor is usually understood* " http://en.wikipedia.org/wiki/Occam%27s_razor

To wit, "the principle recommends selecting the hypothesis that introduces the fewest assumptions and postulates the fewest entities".

But, why would someone write a piece of software to figure the simplest solution? The answer to that question is that there are times when dealing with an overabundance of data – that the answer which would correctly separate the various aspects of that data into categories is not easily discerned.

Everyone has had those moments when the sheer volume of information relating to a problem is so varied that there seems to be no easy way to determine which elements of the information are relevant and which ones are not.

Ross Quinlan's ID3 algorithm, published in the premier edition of *Machine Learning* seeks to resolve all of the conflicting data that is presented to it – IF said data can be resolved into a proper sets of different categories.

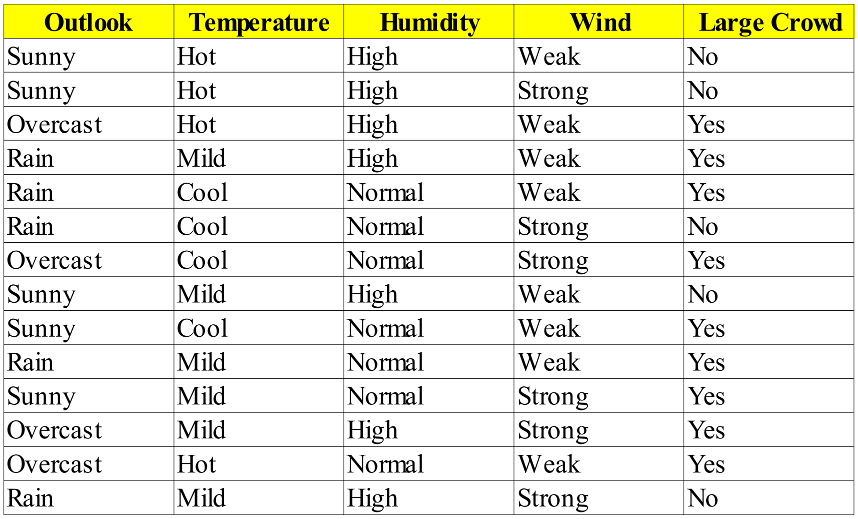
If one were to peruse just the wikipedia entry for the ID3 Algorithm, one sees at least 7 different implementations in languages ranging from the prosaic C# to more exotic Haskell. However, each one of those that are listed operates as a command line program.

ID3A is an attempt by the author to spruce up the presentation of this algorithm into a more "modern" interface so that the end user is not forced to head to the command line and type idiosyncratic codes to make it work as if the year were 1993 instead of 2009.

A Simple Walkthrough

Let us say for the sake of argument that you, dear reader, are the owner of a local community pool. With all of the cut backs going on in the local government you have been asked to adjust your staffing in the most economical means necessary. Since you happened upon knowledge of ID3A you decide to use it to assist you in the development of your staffing schedule.

The first thing you do is spend the next two weeks making notes on various factors relating to the weather and the types of crowds, most especially if they are "large" crowds which would mean you would need to be at full capacity. After two weeks your data looks something like this:



Outlook	Temperature	Humidity	Wind	Large Crowd
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

There are several things to note with regards to the information here and you will need to pay close attention to it when you use ID3A to create your own decision trees.

1. The last column is the "decision", as in the final category. This will always need to be one of two states. In the example above, it is either "yes" or "no"
2. The other columns can have a wider range of options.

So, you load the information into your trusty spreadsheet and save it as a "csv" file (comma separated values) and store it in your folder.

Next, you'll load up the data into ID3 and generate a ruleset. For example, the ruleset that is generated here is as follows:

if Outlook=='Sunny ' and Humidity=='High ' then Large Crowd='No '
elsif Outlook=='Sunny ' and Humidity=='Normal ' then Large Crowd='Yes '
elsif Outlook=='Overcast ' then Large Crowd='Yes '
elsif Outlook=='Rain ' and Wind=='Weak' then Large Crowd='Yes '
elsif Outlook=='Rain ' and Wind=='Strong' and Temperature=='Cool' then Large Crowd='No '
elsif Outlook=='Rain ' and Wind=='Strong' and Temperature=='Mild' then Large Crowd='No'

else raise 'There was not enough information during training to do a proper induction for this data element' end

And even this decision tree is a bit confusing at first blush however, closer inspection shows that there are three states for which there will be large crowds

1. If the Outlook is **Sunny**, the humidity is **Normal**
2. If the Outlook is **Overcast**
3. or if the Outlook is for **Rain** and the wind is **Weak**

According to ID3A, if any of these conditions are met, then there will be a large crowd.

One could go away right here knowing that the decision tree had been built and there would be no further need for examination. And, if the problem was that simple this would be truth of the matter. However, ID3A was not built for problems as simple as this and has an additional function not often presented with implementations of ID3.

ID3A will not only generate the ruleset from the examples given but, it will also allow the user to save that ruleset and come back to it on another day and time and simply load the ruleset and query it, providing the answers from another case and receiving the answer from the decision tree.

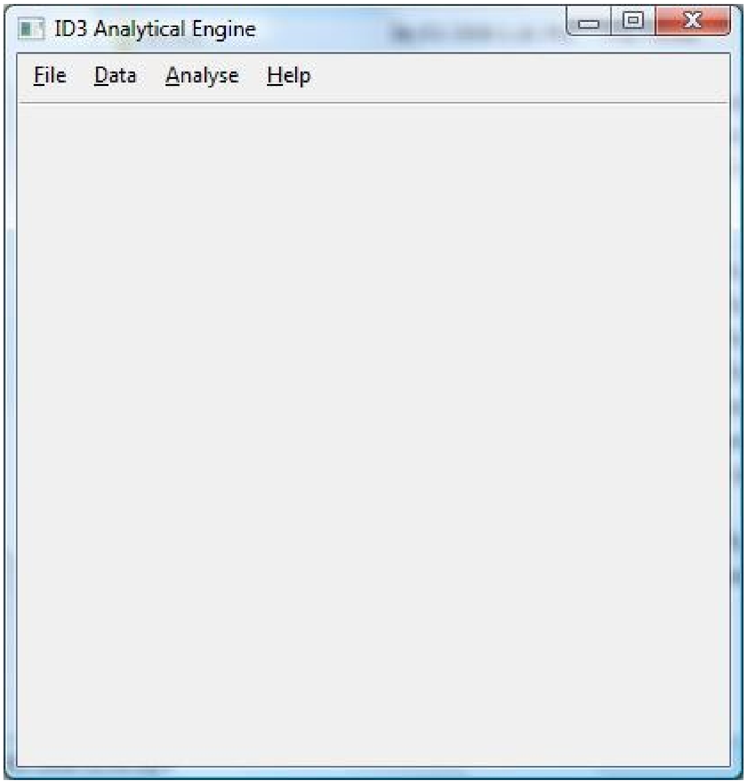
This allows the user to distribute ID3A with a prebuilt ruleset and use it as a functional expert system, providing information based on the pre-generated ruleset without the need for any additional compilation.

Please note that the ID3A ruleset generator is open sourced and will be free of charge however there will likely be a charge for the redistributable application. Contact the author at cognition.crow@gmail.com for further information on licensing of the expert system application.

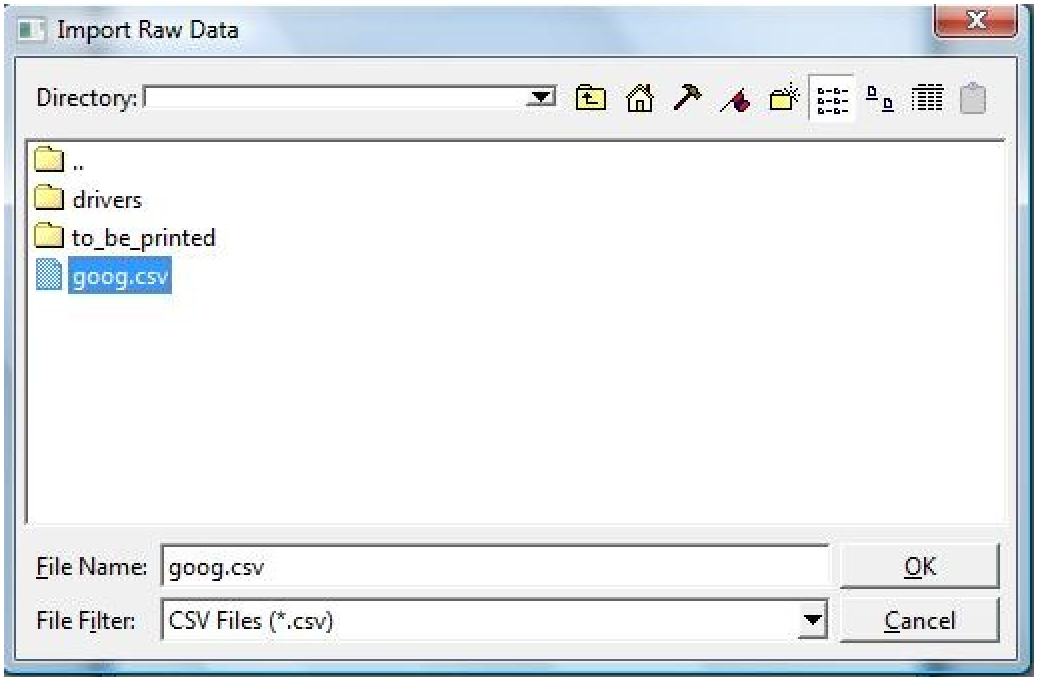
ID3A Usage

What follows below is a step by step usage of the ID3A ruleset generator.

1. Gather your data into a CSV file with the top row being the header and the last column in the data set being the "decision" column. (If this is confusion, refer back to the example for a description of how the data should be laid out)
2. Start the application



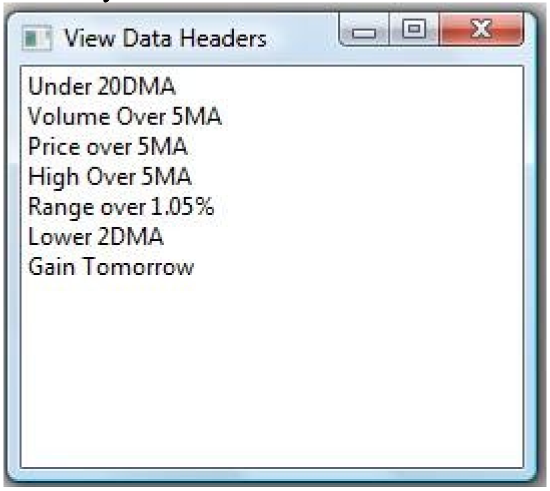
3. Under "File", select "Load CSV" and navigate to the folder where you have stored your data.



File Name: goog.csv

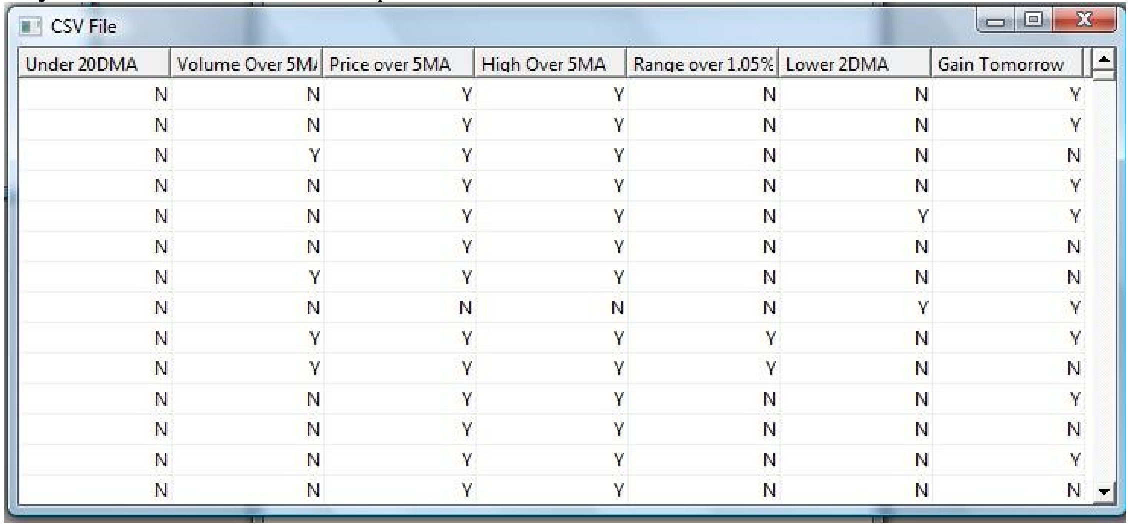
File Filter: CSV Files (*.csv)

4. Once the data has been loaded you will be able to view either the "headers"



Under 20DMA
Volume Over 5MA
Price over 5MA
High Over 5MA
Range over 1.05%
Lower 2DMA
Gain Tomorrow

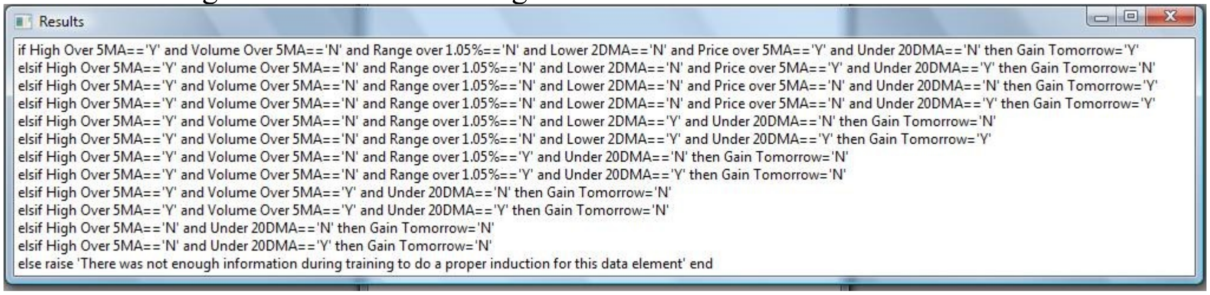
5. Or you will be able to take a quick view at the entire table of data if need be



Under 20DMA	Volume Over 5M/	Price over 5MA	High Over 5MA	Range over 1.05%	Lower 2DMA	Gain Tomorrow
N	N	Y	Y	N	N	Y
N	N	Y	Y	N	N	Y
N	Y	Y	Y	N	N	N
N	N	Y	Y	N	N	Y
N	N	Y	Y	N	Y	Y
N	N	Y	Y	N	N	N
N	Y	Y	Y	N	N	N
N	N	N	N	N	Y	Y
N	Y	Y	Y	Y	N	Y
N	Y	Y	Y	Y	N	N
N	N	Y	Y	N	N	Y
N	N	Y	Y	N	N	N
N	N	Y	Y	N	N	Y
N	N	Y	Y	N	N	N

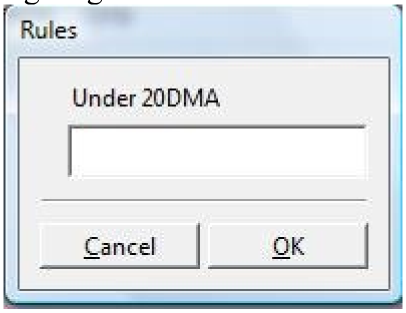
6. Next you will go under the "Analyse" menu item and choose "Generate Rules" and you should

see something similar to the following:



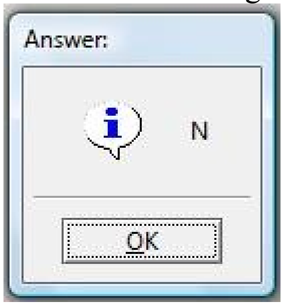
```
if High Over 5MA=='Y' and Volume Over 5MA=='N' and Range over 1.05%=='N' and Lower 2DMA=='N' and Price over 5MA=='Y' and Under 20DMA=='N' then Gain Tomorrow='Y'
elsif High Over 5MA=='Y' and Volume Over 5MA=='N' and Range over 1.05%=='N' and Lower 2DMA=='N' and Price over 5MA=='Y' and Under 20DMA=='Y' then Gain Tomorrow='N'
elsif High Over 5MA=='Y' and Volume Over 5MA=='N' and Range over 1.05%=='N' and Lower 2DMA=='N' and Price over 5MA=='N' and Under 20DMA=='N' then Gain Tomorrow='Y'
elsif High Over 5MA=='Y' and Volume Over 5MA=='N' and Range over 1.05%=='N' and Lower 2DMA=='N' and Price over 5MA=='N' and Under 20DMA=='Y' then Gain Tomorrow='Y'
elsif High Over 5MA=='Y' and Volume Over 5MA=='N' and Range over 1.05%=='N' and Lower 2DMA=='Y' and Under 20DMA=='N' then Gain Tomorrow='N'
elsif High Over 5MA=='Y' and Volume Over 5MA=='N' and Range over 1.05%=='N' and Lower 2DMA=='Y' and Under 20DMA=='Y' then Gain Tomorrow='Y'
elsif High Over 5MA=='Y' and Volume Over 5MA=='N' and Range over 1.05%=='Y' and Under 20DMA=='N' then Gain Tomorrow='N'
elsif High Over 5MA=='Y' and Volume Over 5MA=='N' and Range over 1.05%=='Y' and Under 20DMA=='Y' then Gain Tomorrow='N'
elsif High Over 5MA=='Y' and Volume Over 5MA=='Y' and Under 20DMA=='N' then Gain Tomorrow='N'
elsif High Over 5MA=='Y' and Volume Over 5MA=='Y' and Under 20DMA=='Y' then Gain Tomorrow='N'
elsif High Over 5MA=='N' and Under 20DMA=='N' then Gain Tomorrow='N'
elsif High Over 5MA=='N' and Under 20DMA=='Y' then Gain Tomorrow='N'
else raise 'There was not enough information during training to do a proper induction for this data element' end
```

7. The next step will be to select “Save Ruleset” located under the “File” menu and give the ruleset a name.
8. Once the ruleset has been saved you will be able to query it by selecting “Query” under the “Analyse” menu option and you will be presented with a series of input dialogs which will ask you for information based on the column headings. If at any point there is a mistake in your entry, continue through and begin again.



Under 20DMA

9. After all the elements of the test case have been input via the Query dialog, ID3A will provide an answer that will be in keeping with the ruleset originally given to it:



N

Afterwards, the application can be either re-queried or closed to be launched again when an example is to be tested.

Process for using saved ruleset

1. Launch ID3A
2. Under “File”, select “Load Rules”
3. Under “Analyse”, select “Query”
4. Follow the same steps as listed in 8 and 9 on previous pages.