Assignment 2: Practical Data Mining Project

31005 Advanced Data Analytics
Spring 2018

The goal of this assignment is to develop your skills in a practical data mining project. There are two options.

1. To implement a simple data mining algorithm, ID3, from scratch. Or you could program an algorithm of similar sophistication (seek advice in advance).
2. To do a data mining project. You can use an existing machine learning libraries for the development. We recommend modern Python machine learning libraries such as scikit-learn, pytorch or tensorflow. You can base your development on existing publicly available projects such as repositories downloadable from github.

You can do this assignment by yourself or in pairs, seek advice in advance if more than two people would like to form a team.

For all choices, you need

1. to submit a report on the development of the project (see below for details)
2. and to record a short presentation or screencast highlighting your work (5–10 minutes maximum). The best movies will be shown to the class in the last week.

The main thing is to choose a project that you're interested in and passionate about.

Choice 1: Programming ID3

The first option is to program ID3 using the algorithm described in class. You need to develop software to solve a supervised learning problem (ie. to build a model against a training set), then run the software against a test dataset and report the accuracy of the model. Your program should do the following things:

1. Read a training dataset and a test dataset. The datasets are in the form of text files. See below.
2. Build a model using the training data as input.
3. Print out a representation of the model (ie. the tree or similar).
4. Run the test data against the model, work out the accuracy of the model (ie. How many samples it classified correctly) and print out a confusion matrix to summarise the results.

Format of the Datasets

Please use the Mushroom dataset to test the algorithm implementation. It is a simple comma separated values text file describing whether mushrooms are poisonous or edible. It consists of two parts:

Header line beginning with the comment character "#" followed by the name of each attribute separated by commas. You should ignore any whitespace in the line.

Data lines follow the header. Each data point (or sample point) is on a separate line.

Attribute values are separated by commas. All attributes are categorical and do not have embedded spaces. The attribute named “class” is the class type for the sample and has the value “edible” or “poisonous”. Your task it to build a decision tree that can discriminate between edible and poisonous mushrooms. For more information, see

http://archive.ics.uci.edu/ml/datasets/Mushroom

The Software

You are free to develop the software in any language you like, although C/C++, Java, Python or similar are preferred. If you want to choose a different language, check first with me. The program should be a text–based “console” program. You do not need to worry about a GUI.

What to Submit

There are three parts to your deliverable, which will be around 20 pages.

1. a short description of the design of the program (3 pages maximum);
2. the source code for your program;
3. transcript of output from your program.

It is difficult to give an estimate of the number of lines of code required because it depends on the specific language and the data mining algorithm chosen. However, you should be able to code it all up using 3 or 4 classes in an object–oriented design.

The ID3 algorithm

You should build a decision tree using the **ID3** algorithm given in the 3rd lecture (it is a pretty simple algorithm, feel free to learn it yourself if you choose to start this assignment before Week 3). This algorithm uses the information gain measure to calculate the splits. You should build the decision tree using the training data supplied, then calculate the error on the supplied test/validation data. Since the mushroom dataset is categorical, you will not need to consider the complexities added with real–valued attributes. There is missing data in the mushroom dataset (flagged by “?” values). Don’t treat the missing data specially. Just pretend that “?” is just another value for the attribute in question. Also, do not worry about pruning the tree.

The program must display a text representation of the decision tree. You are free to display the tree in any way you think makes sense, so long as it shows what attributes are tested at each node in the tree. It is acceptable to utilise diagnosis tools provided by machine learning packages for the display of the tree ** as long as the tree is built by your own program, i.e. it is NOT acceptable to form a 2nd tree using the package, and display the 2nd tree directly **.

Hint #1: The trick with building the decision tree is not really the ID3 algorithm which is fairly straightforward. The tricky bit is managing the dataset. Remember that you need to be able to easily split the dataset based on the value of a specific attribute. That means you need to devise a suitable data structure to easily do this split and to work out class frequencies.

Hint #2: Think carefully about the entropy function you need to use when calculating information gain. It’s not quite so simple as in our theoretical discussion. Specifically, what happens when all of the dataset you’re looking at has only one of the two class values? ie. all the mushrooms are edible or all are poisonous? How will you deal with this?

Hint #3: Follow carefully the online learning materials provided Week 3.

Choice 1-alternative: Programming an algorithm of your choice

The second option allows you to choose another algorithm to program, so long as you seek approval from me. One potential method is a multilayer perceptron neural network. You may use a supporting mathematical library to help with the details so long as you code the machine learning algorithm part yourself. Note: It is not acceptable to simply write code to call the Java Weka algorithm or the Python scikit-learn code for the algorithm. I expect you to write the main algorithm yourself. The dataset to be used for the classification (or regression) problem will need to be determined in consultation with me, but as a default we would probably use the mushroom dataset from choice 1 if it makes sense.

Comments about what you need to submit and choice of programming language are as above.

Choice 2: Doing a data mining project

The third choice is to use an existing package to solve a data mining problem. If you want to do this it will not be enough to just use one classification algorithm and copy the output. You need to explore the data, systematically try several algorithms and parameter settings to find the best (by evaluating the quality of the classifiers) and then provide a recommendation.

Format of the Datasets

I'm happy for you to choose a dataset, but check with me first. A very good source is https://www.kaggle.com/datasets

What to Submit

There are three parts to your deliverable.

1. a description of your exploration of the dataset highlighting interesting or important things you found (roughly 5–10 pages with figures);
2. a description of how you approached the problem, which algorithms you looked at and the parameter settings you used (10 pages);
3. your recommended classifier with reasons why (1 page).

Recorded Presentation

Regardless of the choice of project that you do, each group needs to record a short presentation of 5–10 minutes. The idea is to tell the rest of the class about the results of your wonderful project work. You might want to divide your talk into the following general sections:

- Introduction: what problem were you solving?
- Background information about the problem (if needed)
- Your solution.
- Results, possibly including a demonstration if you wrote a program.
- Reflection: what did you learn? what would you do differently next time?

Due Date

Due date 11:59pm 5 Oct 2018.

How to submit Please submit a soft copy on UTS Online. Make sure you put your student number and your name in the document.

Extensions may be granted for assignments after consultation with the Subject Coordinator before the due date.

Late assignments will have 20 percentage points deducted from the total worth to the assignment per day late or part thereof, more than five days late the assignment will receive zero. Special Consideration, for late submission, must be arranged beforehand with the Subject Co–ordinator.

Assessment

Group work This assignment may be done individually or in pairs. Conditions for group work are described in the subject outline. Except for exceptional circumstances (ie. where problems occur in the group), each member will receive the same mark. If there are problems in your group, please see the Subject Coordinator.

Return I will endeavour to return marked assignments within three weeks.

Contribution to final mark This assignment contributes: 40% towards your final mark.

Objectives This assignment supports objectives 1, 3 and 4 and Graduate Attributes C2 and E1 in the subject outline.

Academic Standards Please see the subject outline for details on the ethical standards we expect from you.

Hours An average student should expect to spend around 48 hours to get a 50P result on this assignment.

Code for ID3 and other algorithms may easily be found on the Internet (by you and by me!), but using this code defeats the purpose of the assignment. The point is for you to learn how to solve the program yourselves. Please don't use source code you find on the Internet. I will check all assignments. Also, please don't simply call Weka or scikit-learn code which does the majority of the work from your code.

Marking Scheme

Choice-1 (and alternative)

* Program design (30)
- Data structure: is there discussion on how to store the key status of the model / algorithm? E.g. if the algorithm is ID3, is the discussion on how the tree node represented in memory sufficient? (1-10)
- Overall structure: how the functions / classes are designed and organised? (1-5)
- Interface: clear and implementation independent? (1-5, related to overall structure)
- Efficiency: time and memory efficiency considered?(1-5)

* Clarity and readability (30)
- Variable names are descriptive (1-5)
- Indentation are appropriate (1-5)
- Informative, precise and concise comments (1-5)
- Repeatedly used code blocks are suitably delegated to functions (1-5: related to the structure)
- No goto, reasonably clear execution flow (1-5)
- Assertion or exception handling (1-5)

* Output (20)
- Correctness (1-10)
- Interpretability: the output should be easily understood (1-5)
- Organisation / summarisation: there should be attempt to allow users to find key info if the volume of the output is large (1-5)

* Video Presentation (20)
- Introduction: what problem were you solving? (1-5)
- Background information about the problem and Your solution (1-5).
- Results, possibly including a demonstration if you wrote a program.(1-5)
- Reflection: what did you learn? what would you do differently next time? (1-5)If you choose option 1 or 2 your assignment will be marked based on how well you solve the problem and the design and efficiency of your program.

Choice-2

* Data Exploration (/30)
- Is the business problem understood well and clearly stated in the report? (1-10)
- Is there detailed discussion on the data characteristic related to the problem, such as how the attributes / other aspects may facilitate or pose challenges? (1-10)
- Is there analysis how to deal with / make advantage of the specific data characteristics and therefore to design a solution path? (1-5)
- Clarity in presentation, general logic etc. (1-5)

* Methodology (/30)
- At least two different approaches have been tested, with reasonable motivation, experiment design and criteria for the outcomes (1-10)
- Model selection, (hyper-) parameter tuning scheme for each algorithm (1-10)
- Sophistication of algorithms (1-5, we reward efforts in implementation)
- Clarity in presentation (1-5, consider the tables and figures for results).

* Understanding and recommendation (/20)
- How convincing discussion on the experiment outcomes is, considering both technical soundness and narrative explanation (1-10)
- Does the final recommendation fit the problem (1-10)

* Video Presentation (/20)
- Introduction: what problem were you solving? (1-5)
- Background information about the problem and Your solution (1-5).
- Results, possibly including a demonstration if you wrote a program.(1-5)
- Reflection: what did you learn? what would you do differently next time? (1-5)