Magellan User Report

Clarence Cheung {kcheung6@wisc.edu}
Jin Ruan {jruan3@wisc.edu}

Description

In this report, we summarize some aspects of `Magellan` when our group is working on the project. Due to the limitation of the project, this documentation does not fully capture every aspect of the functionality, but we still hope to provide some feedback based on our experiences. In general, `Magellan` is a user-friendly package in Python. Together with `IPython`, operations and output can be visualized easily and instantly. However, there are still some issues that make this software difficult to use in some situations. We sincerely hope taht this breif report will give some insight for further improvement and expansion.

Blocking

• More functions can be included

Our initial attempt to do blocking is to break Table A into two chunks, based on whether an attribute has or does not have NaN values. Currently, `Magellan` does not support such a function and users need to create their own "MTable" objects. In our case, we consulted the TA and wrote a function that serves our purposes.

• combine_outputs_via_union

Things get a little more complicated when we are merging the tables via the built-in function. The function `combine_outputs_via_union` provided by `Magellan` cannot handle the tables we splitted earlier. In order to do so, the splitted tables must inherit the properties of the original table.

• Debugger

Our group utilized the debugger frequently in our iterative development process. We believe the debugger is very useful in the sense that it tells us the performance

for each blocker. For some reasons, the debugger takes a lot of time to debug regardless of the size of the candidate sets. We believe part of the reason is that the debugger takes the original tables as input, then do a lot of similarity measurement of the tuples from the candidate sets and the original tables. Sometimes it took nearly an hour to debug even though there are hundreds or thousands of tuples in the candidate sets.

- **Suggestions**

It would be appreciated if the future version of `Magellan` can incoporate functions that splits the tables based on certain attributes and their corresponding values. In some of the blocking functions, it would be nice if some estimating time (ETA) can be shown so that the users can distinguish between a running process and a hanged operation. Users normally want to compare different blockers with different similarity functions. As a result, reducing the debugging time gives them more opportunities to do different experiments within the same period of time. If it is possible, the debugger can also allow users to select the preferred similarity functions to debug, rather than using the default values.

Matching

Operations with `Magellan` is smooth in this stage of project and we did not encounter much problem. We would still like to provide some minor feedback though.

- **Labelling**

After randomly sampled tuples from Table C and called the function `label_table`, we did not have any idea that `Magellan` had already opened another window for us to do labeling. It would be appreciated if `Magellan` can quickly switch to the newly opened window so that users are aware of the newly created table. Another potential improvement is that `Magellan` can lock all columns except the label column. While we were doing the labelling, it is possible that the cursors naviagated to other part of the columns and we altered the values of those tuples. The lock of tuples guarantees that the users will not change the data accidentally.

- **Issues with SVM**

For some strange reasons, the numbers displayed by SVM in cross-validations appeared to be zeros. This happened when we included all six classifiers in an `IPython` input cell. However, if each classifier was inserted for each input cell,

the problem went away. Our suggestion is to use each classifier in one `IPython` input cell to mitigate such problem.

- **Trigger function**

In some of our matching experiments, we would like to test with the perfoemance of the trigger functions. Due to the nature of our dataset, sometimes we need to do some union of rules rather than intersecting them. We attempted to write some of these rules, but `Magellan` does not provide such triggers. The documentation is also not comprehensive enough and the examples given in class did not suit our purposes. Nonetheless, our group did not include any rules in the final matching since the rules are complex and they deteriorate the performance.