Open Source IRAP

Implicit Relational Assessment Procedure

License

Version

0.9.6 (5/5/2016)

Written in PsychoPy 1.82

NB This code is still in beta - I haven't used this in an experiment yet or had someone else do a code review.

Notes

Task fidelity

This implimentation of the IRAP has high fidelity to the procedure described in Barnes-Holmes et al. (2010: a sketch of the IRAP and REC model), and to the implimentations of the IRAP written in Visual Basic 6 by Dermot Barnes-Holmes (i.e., the IRAP "2010" etc). Most task parameters are soft coded and can be changed via the `task.xlsx` file.

- Each trial presents a "label" stimulus at the top of the screen, a "target" stimulus in the middle of the screen, and two response option labels at the bottom left and right of the screen. There are two label stimuli categories and two target stimuli categories, which when combined create four "trial types".
- You can employ an arbitrary number of stimulus exemplars per category, but all columns in the excel stimulus file must have the same number of rows (i.e., exemplars).
    - NB This could be changed by dividing the stimuli file into two seperate ones and sampling the labels at random but the targets in a counterbalanced manner. Indeed, this method of unequal selection and counterbalancing of label vs target stimuli is how the original implimentations of the IRAP in Visual Basic 6 by Dermot Barnes-Holmes were constructed (e.g., the "2010" version etc.). However, assuming that there are equal number of label and target stimuli, the current implimentation is technically supperior as it results in an equal number of presentations of all label stimuli exemplars, which the VB6 versions do not.
- The default stimulus file employs text stimuli for both labels and targets. The alternative stimuli file employs images for both. This implimentation can display text or images for either. To change this, edit the stimuli.xlsx file.
    - If using text stimuli, put `blank.png` in the img columns and

the text stimuli in text stimuli column.
    - If using image stimuli, put a single space character (i.e,) in the text stimuli column and the name of the image file (including extension) in the image stimuli column. Failure to do either of these will cause Psychopy to throw an error message.
- The location of the response options (left vs right) can be either fixed or moving randomly. Default is "false" (fixed), set this to "true" for moving.
- Number of trials per block is equal to four times the number of rows in the stimuli.xlsx file. Each block contains an equal number of each trial type.
- Participants complete pairs of blocks in which the response contingencies alternate (e.g., self-positive-true vs. self-positive-false). Each block is preceeded by a customisable responding rule (Rule A and Rule B), and followed by feedback about the median latency and % accuracy in the block.
- The order of presentation of the blocks (i.e., block order) is set for each participant in the dialogue box that appears after you run the task. The default is "a" (rule A first). Set this to "b" for Rule B first. That is, if you're counterbalancing block order, the researcher assign the block order for each participant when the task is run.
- Participant complete practice block pairs (default max 4) until they meet mastery criteria on both blocks in a pair (default median latency <= 2000ms and accuracy >= 80%), and then a fixed number of test block pairs (default 3). If mastery criteria are not bet within the max the task skips the test blocks and goes to the end screen.
- Inter trial interval is set to 400 ms.

Timing accuracy

- PsychoPy is technically capable of better timing accuracy than Visual Basic 6, depending on design choices by the researcher (see Garaizar & Vadillo, 2014).
- The current implimentation is written to be at least as accurate as the VB6 implimentations of the IRAP (i.e., accurate to within a frame or c.17ms). The stimuli to be presented within a block are generated on the preblock rule screen, and then `pop()` 'd on each trial.
- If you're looking for higher accuracy (e.g., for EEG/fMRI work) you'll want to change all timings to frames rather than seconds. You may also want to remove the presentation of images if you're not using them.
    - NB no assessment of jitter has been conducted for the current implimentation.

Usage

- The task is written in PsychoPy, a free and open source python library for delivering psychology experiments.
- To run the task, download a copy of PsychoPy and open either the IRAP.psyexp or IRAP.py file in PsychoPy. PsychoPy runs locally on Windows, Mac, and Linux.
- You can run either the `.psyexp` file or the `.py` file inside PsychoPy. The `.py` file should have greater cross platform support; if you run into errors with the `.psyexp` file use the py instead.
- The escape key quits the task at any time. The return key ends the task properly once it’s complete.
- All stimuli and instructions can be altered by editing the `.xlsx` files. - PsychoPy has unicode support, so translating the task into other languages only requires changes to these excel files. All text presented in the task are set via the excel files.
    - NB a poorly documented bug is that if you zip and unzip

excel files using archieve utility on Mac OS X, unicode characters are no longer correctly displayed and will throw an ASCII error in PsychoPy. Make new excel files to correct the issue.

- When you run the task, an autoresponse 'monkey' can be invoked by setting "UseMonkey" in the dialogue box to "y" or "yes". This will simulate keypresses throughout the task to that you can test your script without you having to hit D and K interminably. NB you may have to lower your accuracy criterion to below 50% (i.e., just put it to 0) for the task to run through entirely, as the monkey simply simulates the D key and then the K key, in that order, on every trial.

Data output and processing

- `.psydat`, `.csv` and `.log` files are produced for each participant. The `.csv` file alone is sufficient to most analyses (e.g., calculation of D scores).
- To my understanding, the format of the `.csv` output files are Tidy Data compliant (Wickam, 2014) and therefore easy to analyse in R with little to no processing needed.
    - NB If a participant gets 100% of trials correct throughout the task then the incorrect response RT column will not be created for that participant. This is a) extremely unlikely, and b) not a problem if you process data files based on column header matching (e.g., most R methods, including the bundled script). However, it can be problematic if your data processing workflow relies on column order rather than column header name (e.g., a SPSS script using a GET command).
- The included `data processing.r` R script produces accuracy and latency summary data as well as D-IRAP scores (overall, trial-type, and split half). Very little familiarity with R is needed: simply

change the set working directory line to the location of your data (e.g., `setwd("~/git/IRAP/data`), and the save output line to your chosen directory (e.g., `write.csv(all_tasks_df, file = '~/git/IRAP/data processing/IRAP_data.csv', row.names=FALSE)`).
    - NB accuracies are calculated by reverse scoring the `feedback_response` variable.

Known issues & to do list

1. R script for processing D scores, development and proofing.
2. Needs a block length multiplier, given that many people will use only 3 or 4 stimulus exemplars. A simple multiplier variable in the task.xlsx file could make this easier.
3. Doesn't seem to deliver test blocks when I last tested it??

Changelog

0.9.6

- Made the shuffle()/pop() code more transparent by writing a function and calling it rather than repeating code. No functional difference for the user, but better code transparency and consistency across the IAT/RRT/IRAP.

0.9.5

1. Added option for block order selection via the dialogue box.
2. Added autoresponse monkey
3. Added option for image stimuli.
4. Added Colored text for pre block rules.

5. Added option for moving response options.
6. Tidied up post block text and locations.
7. Added warning level logging
8. Separated the block layout and stimuli excel files. This allows for an arbitrary number of exemplars for each stimulus category.
9. Commented out the saving of median latency and % accuracy data to the csv file, as it was causing problems in reading the file into R.