Using High Throughput Computing for a Simulation Study on Cross-Validation for Model Evaluation in Psychological Science

Hannah Moshontz, PhD

Background

- Goal pursuit in everyday life
- Machine learning for predicting complex, multiply-determined outcomes

Technical / programming skills

- Advanced training in quantitative methodology
- Proficient w programming languages for statistics — SAS and R
- Prior to this project, very limited BASH, no HTCondor

Project Background

- Lack of norms or guidance on machine learning practices in psychological science
- People incorrectly interpreting / using cross-validated model performance estimates in top journals
- Goal: characterize and give guidance on cross-validated model performance estimates in data contexts typical of psychological science

https://psyarxiv.com/ns9mj/

Simulation Study Overview

Simulation Study Overview

- Entailed running about 144,000 times the number of models in a single ML study
- Within a single ML study, tens of thousands of models are run. Run time is less than a minute to weeks, depending on data context.
- Compute time well over 1 million hours

Simulation Study Overview

- Entailed running about 144,000 times the number of models in a single ML study
- Within a single ML study, tens of thousands of models are run. Run time is less than a minute to weeks, depending on data context.
- Compute time well over 1 million hours

We didn't understand the scale of this project initially.

Our software and scripts

Each job we submitted had:

- R script (.R)
- Arguments (.csv)
- PRE and POST scripts (.sh)
- Submit file (.sub)
- Executable file (.sh)
- (We started with DAGs, but didn't use later)

We used notepad ++, vim, and later created all files in R.

Our software and scripts

Output from each batch

- Zipped outdata - for each job, a summary of the best model performance, plus information about the model (.rds)
- Zipped “job files” - the submit and executable files, the args, the error files, the output files
- The log

How we tracked jobs

nX	nN	UB	Complexity number	Elapsed time range	Memory use (max)	Disk use (max)	Notes
10	100	B	1	5M to 46M	1075	580917	
100	100	B	1	55M to 3H 51M	977	580918	4 holds, released and ran/completed - kendra
1000	100	B	3	2M to 55M	849	580914	3 holds - released/ran with 1 hold - said size of job was 17090, to resolve needed to up the memory
10	500	B	1	11M to 30M	1159	580917	
100	500	B	2	15M to 2H 10M	952	580915	14 jobs held for memory reasons; upped to 2048 - hannah
1000	500	B	3	4M to 4H 10M	1014	580917	
10	1000	B	1	16M to 1H 15M	1612	580917	
100	1000	B	2	24M to 4H 33 M	1076	580915	6 hold, released and ran/complete - kendra
1000	1000	B	3	4M to 5H 13M	1177	580921	
10	100	U	1	7M to 17M	900	580917	
100	100	U	1	52M to 4H 19M	1005	580917	
1000	100	U	3	3M to 47M	865	580914	8 jobs held for fluke memory reasons (impossible peak usage); r
10	500	U	1	8M to 22M	1000	580917	
100	500	U	2	10M to 2H 45 M	946	580916	16 holds - released and ran/completed - kendra
1000	500	U	3	4M to 4H 5M	926	580916	19 holds for fluke memory reasons; all ran fine on release -gayle
10	1000	U	1	9M to 41M	1251	580981	
100	1000	U	2	29M to 8H 28 M	2048	580979	4 held for memory ("Peak usage: 32564 megabytes"). Requested 2200 and they finished without issues. - Hannah
1000	1000	U	3	6M to 9H 45M	1120	580952	8 held for fluke memory reasons (peak usage reported is impossible). - Hannah

Errors and troubleshooting

- Simple errors we made
- Issues related to software when flocking/gliding
- Issues related to errors with our R script
- Issues related to how we broke jobs up / optimization

CHTC staff offered *tremendous* support, both via direct contact and via the excellent HTCondor manual and other online documentation and resources.

Simple errors with HTCondor

- Unix line endings
- Typos in our submit or executables
- Not moving files to the submit server
- Not running PRE.sh
- For big jobs, zipping too many at once

Issues related to flocking, gliding, and R

- Needed to download a support tar (SLIBS) from the SQUID servers
- We changed the version of R we were using, and had issues with package dependencies in our package tar
- There was a set of machines on UW's campus that were having odd issues with base R
- Jobs would get booted when flocking/gliding (many of our jobs were near or just over 8 hours)

Issues related to the project / R scripts

- Adjustments to how we simulated data - reflection and reviewer feedback
- A few contexts had so few positive cases that the models failed, which didn't produce the output that we were expecting

Issues related to optimization

- Making a single zipped file that took hours to unzip, not inspecting contents before unzipping
- We wrote the script in a way that was well-suited to being broken down, but in rigid ways
- We starting running jobs before we had tested the most complex contexts and hit a floor in how simple the jobs could be (the most simple was still taking over 72 hours with a particular algorithm)
- We didn't understand that small efficiencies scale, and are important (e.g., ranger vs RF)

Helper scripts

- Meta-script
  - created all files that needed to run a batch of jobs, including changing line endings, making the R script, making the args file
  - for completed jobs, summarized the log
  - made and ran the other helper scripts
- Check / unzip script
  - checked the outdata and unzipped
  - produced an args file for any missing jobs
- Aggregation script
  - Made data comparable no matter how jobs were broken up

Advice

- Automate what you can to prevent errors & save time
  - Make files and folders descriptive and machine-readable
- Document everything well
  - keep detailed notes about testing and completed jobs
  - save HTC files
- Be mindful of the resources you will use and are using
  - computing hours can be abstract and hard to estimate! check how many hours you / your team has used
  - reevaluate the scope of your project periodically
  - avoid waste through preparation and testing

Advice

- Use the HTCondor manual
  - there are so many useful functions and so much information that you have access to
- Become a pro troubleshooter
  - learn to systematically rule out basic issues and diagnose the issue you have
  - reach out for help with detailed information about what you have done, and with jobids, logs, and other documentation
- Describe the time/resource constraints that informed your research
  - reviewers may not appreciate these constraints unless explained

Closing thoughts

- We couldn’t have conducted this study in my lifetime without HTC
- The CHTC staff are an incredible resource, and this project wouldn’t have been completed without them and the HTCondor manual
- A great training / learning experience re: general programming skills (e.g., BASH, troubleshooting / problem-solving)

Thank you